WORKING PAPER

Computing the distribution: Adaptive finite volume methods for economic models with heterogeneous agents

NORGES BANK RESEARCH

10 | 2019

SEHYOUN AHN

NORGES BANK

Working papers fra Norges Bank, fra 1992/1 til 2009/2 kan bestilles over e-post:
FacilityServices@norges-bank.no

Fra 1999 og senere er publikasjonene tilgjengelige på www.norges-bank.no

Working papers inneholder forskningsarbeider og utredninger som vanligvis ikke har fått sin endelige form. Hensikten er blant annet at forfatteren kan motta kommentarer fra kolleger og andre interesserte. Synspunkter og konklusjoner i arbeidene står for forfatternes regning.

Working papers from Norges Bank, from 1992/1 to 2009/2 can be ordered by e-mail
FacilityServices@norges-bank.no

Working papers from 1999 onwards are available on www.norges-bank.no

Norges Bank's working papers present research projects and reports (not usually in their final form) and are intended inter alia to enable the author to benefit from the comments of colleagues and other interested parties. Views and conclusions expressed in working papers are the responsibility of the authors alone.

ISSN 1502-8190 (online)
ISBN 978-82-8379-101-3 (online)

Computing the Distribution: Adaptive Finite Volume Methods for Economic Models with Heterogeneous Agents

SeHyoun Ahn*†
Norges Bank

June 10, 2019

Abstract

Solving economic models with heterogenous agents requires computing aggregate dynamics consistent with individual behaviors. This paper introduces the finite volume method from the mathematics literature to enlarge the set of numerical methods available to compute dynamics in continuous time. Finite volume discretization methods allow theoretically consistent dimensional and local adaptivity that guarantee the mass conservation and positivity of the distribution function of the discretized system. This paper shows examples of 1) the Ornstein-Uhlenbeck process 2) the Aiyagari-Bewley-Huggett (wealth+income heterogeneity) model and 3) the lifecycle (wealth+income+age heterogeneity) model. The numerical exercises show that for the current dimensionality of the problems in economics, the finite volume method (with or without adaptivity) outperforms pre-existing methods. This paper further provides a companion open-source implementation of the finite volume method at `<https://github.com/sehyoun/adaptive_finite_volume>` to reduce the testing time of the finite volume method.

1 Introduction

With recent advances in computing technology, economists started to model heterogeneous agents explicitly instead of relying on the representative agent assumption. Such analyses follow two steps in general. First, researchers solve the dynamic programming problem to model individual agents' behaviors. Given the behaviors, one solves for the evolution of the distribution of idiosyncratic state variables. A typical strategy is to simulate a finite sample of agents to compute aggregate quantities like the aggregate capital. Though the dynamic programming has received lots of attention, the aggregation step, unfortunately, has not.

The simplicity of implementing the Monte Carlo simulation partly explains the lack of attention. The Monte Carlo simulation method, however, has limitations. As a (stochastic) simulation method, Monte Carlo simulation has sampling errors that only scale at $O(N^{-1/2})$, where N is the number of simulation agents. As documented in Algan et al. (2014), the number of simulation agents to control the sampling error can be quite large. For the example given in the paper, the sampling error masks the underlying macroeconomics even with 100,000 simulation agents. Second, Achdou et al. (2017) shows that one can solve the dynamic programming problem much faster in continuous time. However, the Monte Carlo simulation method does not directly generalize to continuous time, and the generalization introduces many hyperparameters that one needs to tune. The Heun scheme for the Monte Carlo simulation is tested in section 3.2, but the test only reaffirms the limitations of the Monte Carlo method noted in Algan et al. (2014), even after tuning the hyperparameters.

For discrete time, researchers have already started approximating the distribution directly to handle the sampling error. (Algan et al., 2008; Winberry, 2018) use parametric functions, and (Ríos-Rull, 1999; Reiter, 2009; Young, 2010) use histograms to approximate the distribution function. They report speed and

*E-mail: sehyoun.ahn@gmail.com

†This working paper should not be reported as representing the views of Norges Bank. The views expressed are those of the author and do not necessarily reflect those of Norges Bank.

	Monte Carlo Simulation	Variational/Function-Approximation	
		Parametric	Non-Parametric
Discrete Time	Krusell & Smith (1998)	Algan et al. (2008) Winberry (2018)	Ríos-Rull (1999) Reiter (2009) Young (2010)
Continuous Time	Euler-Maruyama[4] Heun Method		Achdou et al. (2017) This Paper

Table 1: Aggregation methods in economics.

accuracy gains over the Monte Carlo simulation. In fact, the function approximation method can also be coupled with the perturbation method to solve economic problems with aggregate shocks (See Reiter (2009) for discrete time and Ahn et al. (2018) for continuous time). The perturbation method in continuous time, shown in Ahn et al. (2018), is faster and more accurate than other alternatives in the JEDC special issue on heterogeneous agent methods (Den Haan, 2010).

This paper is a continuation of the "histogram" approach to continuous time. Though unrecognized in economics, this approach has been around for decades in the mathematics literature on discretization of partial differential equations. In a groundbreaking work, Achdou et al. (2017) uses the finite *difference* methods to solve for the distribution in continuous time.[1] The advantages of the (non-parametric) function approximation approach explain some of the speed gains exhibited in (Achdou et al., 2017; Ahn et al., 2018). However, the finite difference method has some limitations, and hence, this paper introduces the finite *volume* method from the mathematics/engineering literature to address these issues.

The need to preserve mass and to ensure positive density restricts the finite difference method to uniform regular grids (see figure 1a). To handle this difficulty, Achdou et al. (2017) introduces a numerical patch (not yet justified by theory) to be used with non-uniform regular grids (see figure 1b). The finite volume method, however, converges to the correct distribution even with non-uniform regular grids (Theorem 3.8 of Eymard et al. (2000)).[2] In fact, the finite volume method is flexible enough to allow non-regular grids like the one shown in figure 1c. Researchers have successfully leveraged the flexibility of the finite volume method with adaptive refinements to solve models faster and/or more accurately in molecular biology (Ferm et al., 2006), physics (Lan et al., 2002), semi-conductor design (Li et al., 2001), and tsunami/flood dynamics (D. L. George & LeVeque, 2006; D. George, 2011).[3] For economic problems, the adaptive refinements introduced here can lead to an accuracy gain over regular grids, and the speed gain is confirmed for test problems in section 3, the same accuracy can be achieved using only 14% of the grid points required by the regular grid for the lifecycle model.

Before proceeding to the rest of the paper, an open-sourced code/toolbox for this paper is available at `<https://github.com/sehyoun/adaptive_finite_volume>` with the corresponding documentation at `<https://sehyoun.com/adaptive_finite_volume>`. Applied researchers are recommended to skim over section 2.1 and work through the tutorials in the documentation.

2 Finite Volume Method

2.1 Intuition

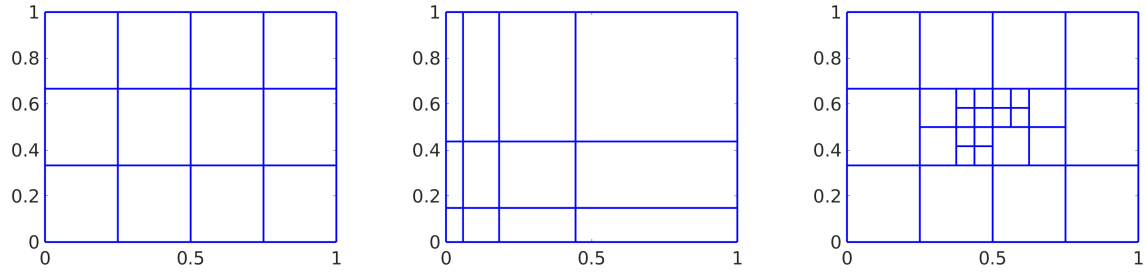
This section introduces the intuition behind the finite volume method. We recommend that most readers read the intuition and follow the tutorials in the documentation. Readers only interested in the formal derivation can jump to section 2.2.

[1] In continuous time, this step solves the Fokker-Planck equation.

[2] With a uniform regular grid, the finite difference and finite volume method result in the same discretized system.

[3] Grids with this type of local adaptivity do not satisfy the regularity condition of theorem 3.8 cited above. However, an adjusted convergence proof is required and referenced in section 3.5.2 of Eymard et al. (2000).

[4] These are standard methods that do not have references in economics. See Rüemelin (1982) for a comparison of the different approximation methods for Monte Carlo simulations.



(a) Uniformly-spaced regular grid (b) Non-uniform regular grid (c) Adaptive grid

Figure 1: Different type of grids: Finite difference method only guarantees the positivity and the conservation of mass under the uniformly-spaced regular grid.

When one approximates distributions with parametric/non-parametric functions, the biggest difficulty comes from the computation of the dynamics from the given (approximate) distribution to the resulting (approximate) distribution consistent with individual behaviors. In the discrete time literature, Ríos-Rull (1999) and Young (2010) compute the dynamics using different sets of interpolations for the histograms. The finite volume method can be considered as an extension of this approach to continuous time.[5] However, as an advantage of continuous time, the finite volume method results in a more intuitive accounting exercise, where one accounts for flows between cells instead of relying on interpolations. As an example, consider the wealth dynamics of a household given by

$$\mathrm{d}x = s(x)\,\mathrm{d}t + \sigma\,\mathrm{d}W_t \quad t \in [0,\infty), x \in \mathbb{R} \tag{1}$$

where $s(x)$ is the savings behavior of households and $\sigma\,\mathrm{d}W_t$ the (idiosyncratic) exogenous changes in wealth level with W_t being a Brownian motion. The distribution of households consistent with the given dynamics is a function over $\mathbb{R}$ and is approximated by a discretization of splitting the domain into cells with boundaries $x_1, \ldots, x_{n+1}$. For notation, denote the total mass of households in a given cell, $[x_i, x_{i+1}]$, by G_i. Hence,

$$G_i = \int_{x_i}^{x_{i+1}} g(x)\,\mathrm{d}x. \tag{2}$$

where $g(x)$ is the density function.

To translate the behavior given in equation (1) to a discretized system, first consider the savings behavior, $s(x)$. The diagram of the behavior is given in figure 2. Without loss of generality, 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. Hence, households will move from the i^{th} cell to the $(i+1)^{\text{th}}$ cell (colored red in the figure) if and only if they were in the range $[x_{i+1} - s\Delta t, x_{i+1}]$ (shaded yellow 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} cell to the $(i+1)^{\text{th}}$ cell. With the assumption of uniform distribution within the cell, i.e., $g_i \approx \dfrac{G_i}{x_{i+1} - x_i}$, one ends up with the dynamics of

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

Before proceeding, note that as households flow from cell i to cell $(i+1)$, the total mass of households is naturally preserved. Also, the flow out of cell i is non-zero only if $G_{i,t}$ is positive. Hence, the positivity

[5] Due to the timing of the literature, (Ríos-Rull, 1999; Young, 2010) should, in fact, be considered a discrete time application/approximation of the finite volume method.

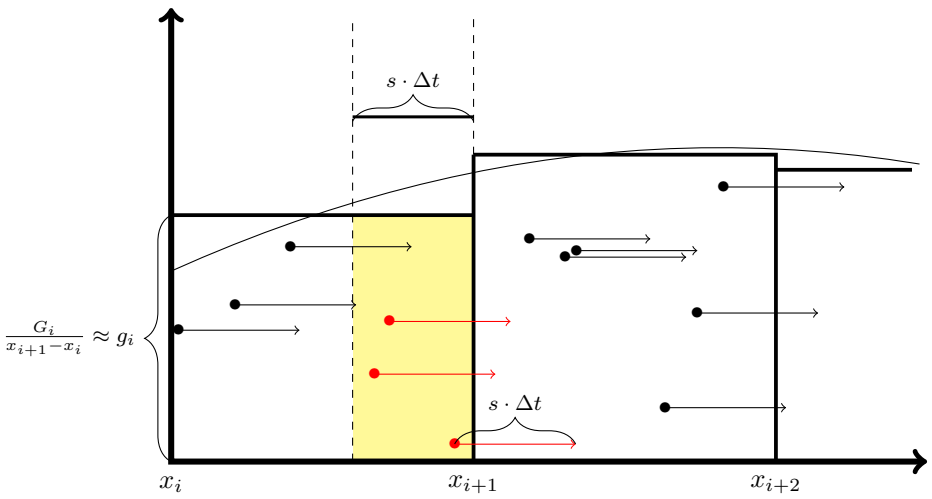


Figure 2: As particles flow at the rate s, particles will flow $s \cdot \Delta t$ (shown with arrows) in the next Δt time period. Hence, the particles within $s \cdot \Delta t$ from the boundary flow from the left cell to the right. Hence, a total of $g_i \cdot s \cdot \Delta t$ of particles (the area shaded yellow) flows into the right cell over Δt time period.

of the distribution is preserved. This, again, is not guaranteed with the finite difference method or other parametric methods.

A similar accounting exercise can be conducted for the diffusion term. The diagram for the diffusion term is shown in figure 3. With the diffusion, the deviation from the previous values, denoted by arrows in the figure, is normally distributed around x_{prev} with the variance of $\sigma^2 \Delta t$. In the figure, as the red colored particles move to the left cell and the green-colored particles to the right, one has to compute the net difference of the red and green to account for the net flow between the cells. Since the Gaussian distribution is symmetric, the difference in the number of particles currently in the cells determines the direction and size of the net flow across the two cells. As shown in figure 3, if the slope of the distribution function is given by $\dfrac{\mathrm{d}g}{\mathrm{d}x}$, then the difference between i^{th} cell and $(i-1)^{\text{th}}$ cell over a one-standard deviation is given by the area of the yellow shaded triangle, $\dfrac{\mathrm{d}g}{\mathrm{d}x}\sigma^2 \Delta t$. The particles will flow from the right cell to the left at a rate proportional to the size of the yellow triangle. This is not the exact value as the probability of crossing the boundary decreases the further away the household is from the cell boundary. A more precise description/explanation will be given in the next section, but letting C be the proper scale parameter for now, the resulting discretized system is

$$G_{i+1,t+\Delta t} = G_{i+1,t} + \frac{G_{i,t}}{x_{i+1}-x_i} s(x_{i+1})\Delta t - C \left.\frac{\mathrm{d}g}{\mathrm{d}x}\right|_{x_{i+1}} \sigma^2 \Delta t + O(\Delta t^2)$$

$$G_{i,t+\Delta t} = G_{i,t} - \frac{G_{i,t}}{x_{i+1}-x_i} s(x_{i+1})\Delta t + C \left.\frac{\mathrm{d}g}{\mathrm{d}x}\right|_{x_{i+1}} \sigma^2 \Delta t + O(\Delta t^2)$$

Making the uniform distribution within cell assumption again, $\dfrac{\mathrm{d}g}{\mathrm{d}x}$ can be approximated by

$$\frac{\mathrm{d}g}{\mathrm{d}x} \approx \frac{\frac{G_i}{x_{i+1}-x_i} - \frac{G_{i-1}}{x_i - x_{i-1}}}{\frac{x_{i+1}-x_{i-1}}{2}}$$

resulting in a discretized system only in terms of G_i's and x_i's. Again, the flow between cells is always accounted for and the outflow is positive only if the cell has positive mass. Hence, the conservation of mass and the positivity of distribution hold even with diffusion terms.

It is impossible to summarize the entire literature on the finite volume method in one paper; hence, in this paper, only the finite-volume discretization scheme with the upwind approximation (defined later)

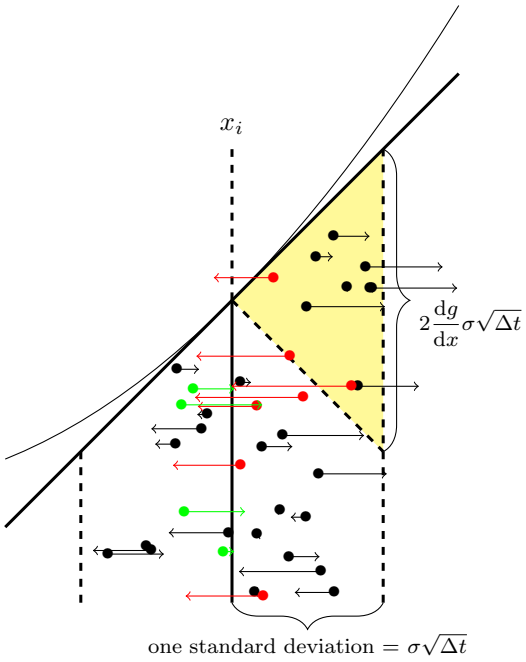


Figure 3: For diffusion, particles move randomly from the initial point distributed as $N(0, \sigma\sqrt{\Delta t})$. These movements are denoted by arrows. Given this randomness, particles can move from the left cell to the right (colored green) and right to left (colored red). For a given particle the same distance away from the boundary, the probability of moving to the respective cell is the same from the symmetry of the normal distribution $N(0, \sigma\sqrt{\Delta t})$. Hence, the *net* flow between two cells will depend only on the difference between the number of particles between the two cells. In the diagram, given the slope of $\frac{\mathrm{d}g}{\mathrm{d}x}$, i^{th} cell has more particles corresponding to the yellow-shaded region (draw based on one standard deviation) as the non-shaded area would cancel, as mentioned above. Therefore, the flow from the i^{th} cell to $(i-1)^{th}$ cell will be proportional to $\frac{\mathrm{d}g}{\mathrm{d}x}\sigma^2\Delta t$.

is provided for its intuitiveness. The derivations are presented in the next two subsections, but one can jump to the discretized dynamics given in equation (6) for one dimension, and equation (11) for higher dimensions. Lastly, though not introduced in this paper for simplicity of discourse, there are many possible discretization schemes within the finite volume methods, e.g., upwind scheme and centered difference scheme. The modified upwind scheme of Axelsson & Gustafsson (1979) is recommended for most applications. The modified upwind method is more accurate than the (simple) upwind discretization, but is more robust to many applications compared to the center-difference method. The modified upwind method is available in the open-sourced codes/toolbox and is used for the numerical experiments in section 3.

2.2 One Dimension

Given the behaviors of households, the equation for the distribution (in continuous time) can be written as a Fokker-Planck-Kolmogorov equation:

$$\frac{\mathrm{d}g}{\mathrm{d}t} = -\frac{\mathrm{d}}{\mathrm{d}x}\left(s(x)\cdot g(x)\right) + \nu\frac{\mathrm{d}^2 g}{\mathrm{d}x^2} \tag{3}$$

where $g(\cdot)$ is the distribution function.[6]

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

[6]Refer to Achdou et al. (2017) for an intuitive introduction.

$$\vec{G} = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{bmatrix}$$

where G_i's are the mass in the cell $[x_i, x_{i+1}]$ given by

$$G_i = \int_{x_i}^{x_{i+1}} g(x)\,\mathrm{d}x \tag{4}$$

Dynamics of the $\vec{G}$ can be computed using equation (3) and (4):

$$\begin{aligned}
\frac{\mathrm{d}G_i}{\mathrm{d}t} &= \int_{x_i}^{x_{i+1}} \frac{\mathrm{d}g}{\mathrm{d}t}\,\mathrm{d}x \\
&= \int_{x_i}^{x_{i+1}} -\frac{\mathrm{d}}{\mathrm{d}x}\left(s(x)\cdot g(x)\right) + \nu\frac{\mathrm{d}^2 g}{\mathrm{d}x^2}\,\mathrm{d}x \\
&= -\left(s(x)\cdot g(x)\right)\Big|_{x_i}^{x_{i+1}} + \nu\frac{\mathrm{d}g}{\mathrm{d}x}\Big|_{x_i}^{x_{i+1}} \\
&= -\left(s(x_{i+1})\cdot g(x_{i+1})\right) + \left(s(x_i)\cdot g(x_i)\right) + \nu\frac{\mathrm{d}g}{\mathrm{d}x}(x_{i+1}) - \nu\frac{\mathrm{d}g}{\mathrm{d}x}(x_i)
\end{aligned} \tag{5}$$

Before proceeding, equation (5) shows a reason to use the finite volume discretization. Any increase of one cell either from $s \cdot g$ or $\dfrac{\mathrm{d}g}{\mathrm{d}x}$ term is from a decrease of the same magnitude in a neighboring cell. Hence, the discretization is guaranteed to preserve mass.

To finish the discretization, since only $\{G_i\}$'s and $\{x_i\}$'s are allowed in the discretized system, $s \cdot g$ and $\dfrac{\mathrm{d}g}{\mathrm{d}x}$ need to be approximated in terms of $\{G_i\}$'s and $\{x_i\}$'s. There are many different approximations in the mathematics literature, and different approximations will lead to differences in accuracies. However, there is one very natural approximation. Suppose that the "$g(x)$" is assumed to be constant within a cell. Then

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

As the size of the cell decreases, the approximation error will decrease. A similar approximation can be made with $\frac{\mathrm{d}g}{\mathrm{d}x}$ assuming constant values within the cell and applying a finite difference approximation:

$$\begin{aligned}
\frac{\mathrm{d}g}{\mathrm{d}x}\Big|_{x_i} &\approx \frac{g\left(\frac{x_i + x_{i+1}}{2}\right) - g\left(\frac{x_{i-1} + x_i}{2}\right)}{\frac{x_{i+1} - x_{i-1}}{2}} \\
&= \frac{\frac{G_i}{x_{i+1} - x_i} - \frac{G_{i-1}}{x_i - x_{i-1}}}{\frac{x_{i+1} - x_{i-1}}{2}}
\end{aligned}$$

Following the upwind scheme of choosing the $g(x)$ from the direction of the flow to ensure positivity of the

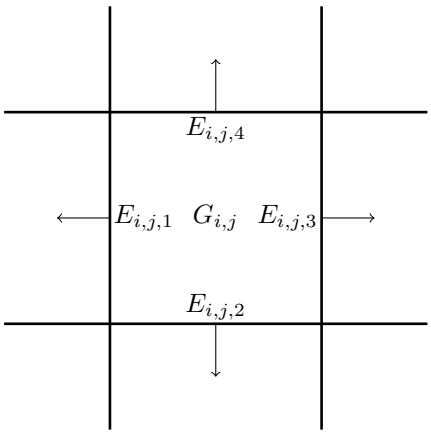


Figure 4: Regular Cell

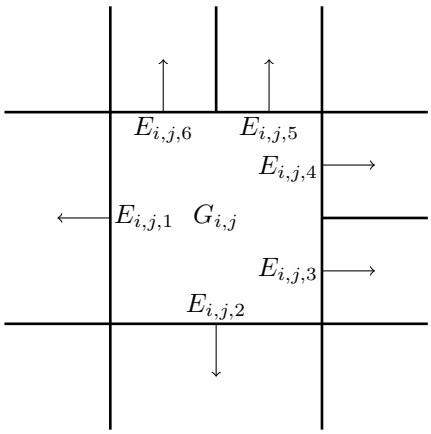


Figure 5: Cell with Adapted Neighbors

distribution[7] and putting above approximations together, the final discretization equation is:

$$\begin{aligned}\frac{\mathrm{d}G_i}{\mathrm{d}t} =& -\left[s(x_{i+1})\cdot\left(\frac{G_{i+1}}{x_{i+2}-x_{i+1}}\cdot\mathbb{1}(s(x_{i+1})<0)+\frac{G_i}{x_{i+1}-x_i}\cdot\mathbb{1}(s(x_{i+1})>0)\right)\right] \\ &+\left[s(x_i)\cdot\left(\frac{G_i}{x_{i+1}-x_i}\cdot\mathbb{1}(s(x_i)<0)+\frac{G_{i-1}}{x_i-x_{i-1}}\cdot\mathbb{1}(s(x_i)>0)\right)\right] \\ &+\nu\left[\frac{2G_{i+1}}{(x_{i+2}-x_{i+1})\cdot(x_{i+2}-x_i)}-\frac{2G_i}{(x_{i+1}-x_i)\cdot(x_{i+2}-x_i)}\right] \\ &-\nu\left[\frac{2G_i}{(x_{i+1}-x_i)\cdot(x_{i+1}-x_{i-1})}-\frac{2G_{i-1}}{(x_i-x_{i-1})\cdot(x_{i+1}-x_{i-1})}\right]\end{aligned} \tag{6}$$

Though seemingly messy, each element of equation (6) simply accounts for flows between cells from the drift or the diffusion. Also, the expression is linear in $\vec{G}$, which results in a simple matrix equation of

$$\frac{\mathrm{d}\vec{G}}{\mathrm{d}t} = A\vec{G} \tag{7}$$

where A is the matrix representation of equation (6).

Equation (7) highlights two advantages of the finite volume method.[8] First, the linearity of equation (7) allows one to leverage the literature on numerical linear algebra. For example, solving for the steady state just requires solving a linear system $A\vec{G} = 0$, which has a very fast solution method. Second, equation (7) separates the drift/diffusion dependent part of the equation from the distribution dependent part through a matrix multiplication. This simplifies the implementation of the finite volume method. If a program constructs the matrix A given the grid, drift and diffusion, 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. This is the approach taken with the open-sourced code where a grid class generates the matrix A given the grid, the drift terms, and the diffusion terms.

2.3 Higher Dimension

The same concept can be generalized to higher dimensional Fokker-Planck equations. In one-dimension, the only possible way of discretizing the space was to use histograms/intervals. However, in higher dimensions,

[7] This is only first-order accurate, and the author recommends using the modified upwind scheme introduced in Axelsson & Gustafsson (1979), where the diffusion term is adjusted to increase accuracy. The implementation is provided at `<https://github.com/sehyoun/adaptive_finite_volume>`. The center difference scheme is also available there. The scheme to use depends on the application, but for most applications, the modified upwind scheme strikes the best balance between accuracy and generality.

[8] These advantages are shared with the finite difference method, and can thus be considered the advantages of the continuous time approach.

there are many ways to discretize the space. In this paper, however, only rectangular grids, where edges are parallel to a coordinate direction as shown in figure 1b and figure 1c, will be considered. Taking one cell from a two-dimensional grid, the flows from the discretized cell are visualized in figure 4.[9] Again, consider the Fokker-Planck-Kolmogorov equation in n-dimension:

$$\frac{\mathrm{d}g}{\mathrm{d}t} = -\sum_{k=1}^{n} \frac{\mathrm{d}}{\mathrm{d}x_j} \left(s_j(\boldsymbol{x}) \cdot g(\boldsymbol{x})\right) + \sum_{k=1}^{n} \nu_j \frac{\mathrm{d}^2 g}{\mathrm{d}x_j^2}(\boldsymbol{x}) \tag{8}$$

Before proceeding, for simplicity let $\boldsymbol{i} = (i_1, i_2, \ldots, i_n)$ be a multi-index, and $-\boldsymbol{i}_k = (i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_n)$. As in one dimension, the total mass in a given cell is used for the discretization, i.e.,

$$G_{\boldsymbol{i}} = G_{i_1, i_2, \ldots, i_n} = \int_{G_{\boldsymbol{i}}} \cdots \int g(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \int_{x_{n,i_n}}^{x_{n,i_n+1}} \cdots \int_{x_{1,i_1}}^{x_{1,i_1+1}} g(\boldsymbol{x}) \, \mathrm{d}x_1 \ldots \mathrm{d}x_n \tag{9}$$

Using equation (8) and equation (9), one gets

$$\begin{aligned}
\frac{\mathrm{d}G_{\boldsymbol{i}}}{\mathrm{d}t} &= \int_{G_{\boldsymbol{i}}} \cdots \int \frac{\mathrm{d}g}{\mathrm{d}t} \, \mathrm{d}\boldsymbol{x} \\
&= \int_{G_{\boldsymbol{i}}} \cdots \int \sum_{k=1}^{n} \left[(-\frac{\mathrm{d}}{\mathrm{d}x_k} \left(s_k(\boldsymbol{x}) \cdot g(\boldsymbol{x})\right) + \nu_k \frac{\mathrm{d}^2 g}{\mathrm{d}x_k^2} \right] \mathrm{d}\boldsymbol{x} \\
&= \sum_{k=1}^{n} \int_{G_{\boldsymbol{i}}} \cdots \int \left[-\frac{\mathrm{d}}{\mathrm{d}x_k} \left(s_k(\boldsymbol{x}) \cdot g(\boldsymbol{x})\right) + \nu \frac{\mathrm{d}^2 g}{\mathrm{d}x_k^2} \right] \mathrm{d}\boldsymbol{x} \\
&= \int_{G_{-\boldsymbol{i}_1}} \cdots \int \left[-\left(s_1(\boldsymbol{x}) \cdot g(\boldsymbol{x})\right)\big|_{x_{1,i_1}}^{x_{1,i_1+1}} + \nu \frac{\mathrm{d}g}{\mathrm{d}x_1}\bigg|_{x_{1,i_1}}^{x_{1,i_1+1}} \right] \mathrm{d}x_{-1} + \sum_{k=2}^{n} \cdots \\
&= \int_{G_{-\boldsymbol{i}_1}} \cdots \int -s_1(x_{1,i_1+1}, x_{-\boldsymbol{i}_1}) \cdot g(x_{1,i_1+1}, x_{-\boldsymbol{i}_1}) + s_1(x_{1,i_1}, x_{-\boldsymbol{i}_1}) \cdot g(x_{1,i_1}, x_{-\boldsymbol{i}_1}) \\
&\qquad + \nu \frac{\mathrm{d}g}{\mathrm{d}x_1}(x_{1,i_1+1}, x_{-\boldsymbol{i}_1}) - \nu \frac{\mathrm{d}g}{\mathrm{d}x_1}(x_{1,i_1}, x_{-\boldsymbol{i}_1}) \, \mathrm{d}x_{-1} + \sum_{i=2}^{n} \cdots \\
&= \int_{G_{-\boldsymbol{i}_1}} \cdots \int \left[-s_1(x_{1,i_1+1}, x_{-\boldsymbol{i}_1}) \cdot g(x_{1,i_1+1}, x_{-\boldsymbol{i}_1}) + \nu \frac{\mathrm{d}g}{\mathrm{d}x_1}(x_{1,i_1+1}, x_{-\boldsymbol{i}_1}) \right] \mathrm{d}x_{-1} \\
&\qquad + \int_{G_{-\boldsymbol{i}_1}} \cdots \int -1 \cdot \left[-s_1(x_{1,i_1}, x_{-\boldsymbol{i}_1}) \cdot g(x_{1,i_1}, x_{-\boldsymbol{i}_1}) + \nu \frac{\mathrm{d}g}{\mathrm{d}x_1}(x_{1,i_1}, x_{-\boldsymbol{i}_1}) \right] \mathrm{d}x_{-1} \\
&\qquad + \sum_{i=2}^{n} \cdots \\
&= \sum_{E \in \text{edges}} \int_{E} \cdots \int (1 - 2 \cdot \chi_{\text{lower}}(E)) \left[-s_E(\boldsymbol{x}) \cdot g(\boldsymbol{x}) + \nu \frac{\mathrm{d}g}{\mathrm{d}x_E}(\boldsymbol{x}) \right] \mathrm{d}x_{-\boldsymbol{i}}
\end{aligned} \tag{10}$$

where $\chi_{\text{lower}}(\cdot)$ is an indicator function that is 1 if the edge is lower and 0 if upper and the subscript E denotes the coordinate direction of the edge.[10]

[9] The calculation given below can be generalized to cells with non-parallel edges by computing the dot product between the flows and the normal direction of the edge. This is an application of Stokes' theorem, but the dot product is not necessary when the edges are parallel.

[10] For example, in figure 4, for the cell $G_{i,j}$, $\chi_{\text{lower}}(E_{i,j,1}) = 1$, $\chi_{\text{lower}}(E_{i,j,2}) = 1$, $\chi_{\text{lower}}(E_{i,j,3}) = 0$, and $\chi_{\text{lower}}(E_{i,j,4}) = 0$. In mathematics, this idea is summarized by the "normal direction" of the surface. The concept is not introduced in this paper for simplicity.

The discretization given in equation (10) is the same as one dimension except that the cell boundaries are $\mathbb{R}^{n-1}$-dimensional hyperplanes and one needs to compute an integral over the hyperplane to get the total flow from one cell to the other. One can make a constant-within-cell approximation to get,

$$\begin{aligned}
&\int_E \cdots \int (1 - 2 \cdot \chi_{\text{lower}}(E)) \left[-s_E(\boldsymbol{x}) \cdot g(\boldsymbol{x}) + \nu \frac{\mathrm{d}g}{\mathrm{d}x_E}(\boldsymbol{x}) \right] \mathrm{d}x_{-\boldsymbol{i}} \\
&\qquad = (1 - 2 \cdot \chi_{\text{lower}}(E)) \left[-s_E(\tilde{\boldsymbol{x}}) \cdot g(\tilde{\boldsymbol{x}}) + \nu \frac{\mathrm{d}g}{\mathrm{d}x_E}(\tilde{\boldsymbol{x}}) \right] \int_E \cdots \int 1 \, \mathrm{d}x_{-\boldsymbol{i}} \\
&\qquad = \underbrace{(1 - 2 \cdot \chi_{\text{lower}}(E))}_{\text{flow direction}} \underbrace{\left[-s_E(\tilde{\boldsymbol{x}}) \cdot g(\tilde{\boldsymbol{x}}) + \nu \frac{\mathrm{d}g}{\mathrm{d}x_E}(\tilde{\boldsymbol{x}}) \right]}_{\text{flow rate at the edge}} \cdot \underbrace{m(E)}_{\text{area}}
\end{aligned}$$

where $\widetilde{\boldsymbol{x}} \in E$ is a point to approximate the function values within the edge, and $m(E)$ is the area of the surface. This expression has a very intuitive interpretation. The rate of increase of the mass of a given cell is determined by the direction and flow rate at the boundary multiplied by the size of the boundary. In other words, the flow through a tube is the product of the speed of the water and the size of the tube. This product does not exist in the one-dimensional case, but exists in all other dimensions.

Finally, one can make the same approximation for $s(\boldsymbol{x}) \cdot g(\boldsymbol{x})$ and $\left.\frac{\mathrm{d}g}{\mathrm{d}x}\right|_{\boldsymbol{x}}$ as in one dimension, and end up with the discretized system. For simplicity of notation, let $\boldsymbol{i}|_{\boldsymbol{E}}$ be the index for the cell across the edge E,[11] then

$$\begin{aligned}
\frac{\mathrm{d}G_{\boldsymbol{i}}}{\mathrm{d}t} &= \sum_{E \in \text{edges}} (1 - 2 \cdot \chi_{\text{lower}}(E)) \left[-s_E(\tilde{\boldsymbol{x}}) \cdot g(\tilde{\boldsymbol{x}}) + \nu \frac{\mathrm{d}g}{\mathrm{d}x_E}(\tilde{\boldsymbol{x}}) \right] \cdot m(E) \\
&= \sum_{E \in \text{lower edges}} (-1) \cdot \left[-s_E(\tilde{\boldsymbol{x}}) \cdot g(\tilde{\boldsymbol{x}}) + \nu \frac{\mathrm{d}g}{\mathrm{d}x_E}(\tilde{\boldsymbol{x}}) \right] \cdot m(E) \\
&\qquad + \sum_{E \in \text{upper edges}} \left[-s_E(\tilde{\boldsymbol{x}}) \cdot g(\tilde{\boldsymbol{x}}) + \nu \frac{\mathrm{d}g}{\mathrm{d}x_E}(\tilde{\boldsymbol{x}}) \right] \cdot m(E) \\
&= \sum_{E \in \text{lower edges}} \left[s_E(\tilde{\boldsymbol{x}}) \cdot \left(\frac{G_{\boldsymbol{i}}}{m(\boldsymbol{i})} \cdot \mathbb{1}(s_E(\tilde{\boldsymbol{x}}) < 0) + \frac{G_{\boldsymbol{i}|_{\boldsymbol{E}}}}{m(\boldsymbol{i}|_{\boldsymbol{E}})} \cdot \mathbb{1}(s_E(\tilde{\boldsymbol{x}}) > 0) \right) - \nu \frac{\mathrm{d}g}{\mathrm{d}x_E}(\tilde{\boldsymbol{x}}) \right] \cdot m(E) \qquad (11) \\
&\quad + \sum_{E \in \text{upper edges}} \left[-s_E(\tilde{\boldsymbol{x}}) \cdot \left(\frac{G_{\boldsymbol{i}}}{m(\boldsymbol{i})} \cdot \mathbb{1}(s_E(\tilde{\boldsymbol{x}}) > 0) + \frac{G_{\boldsymbol{i}|_{\boldsymbol{E}}}}{m(\boldsymbol{i}|_{\boldsymbol{E}})} \cdot \mathbb{1}(s_E(\tilde{\boldsymbol{x}}) < 0) \right) + \nu \frac{\mathrm{d}g}{\mathrm{d}x_E}(\tilde{\boldsymbol{x}}) \right] \cdot m(E)
\end{aligned}$$

where

$$\frac{\mathrm{d}g}{\mathrm{d}x}(\tilde{\boldsymbol{x}}) \approx \frac{\left(\pm \frac{G_{\boldsymbol{i}|_{\boldsymbol{E}}}}{m(\boldsymbol{i}|_{\boldsymbol{E}})} \mp \frac{G_{\boldsymbol{i}}}{m(\boldsymbol{i})} \right)}{\Delta x}$$

Theorem 3.8 from Eymard et al. (2000) applies, the discretized solution approaches the true solution with finer grid points. This seemingly daunting expression just accounts for drifts and diffusions through edges. The expression is, again, linear in $\vec{G}$ and can be written as

$$\frac{\mathrm{d}\vec{G}}{\mathrm{d}t} = A\vec{G} \qquad (12)$$

with all the benefits noted in section 2.2 carrying over. To reiterate, this means that applied researchers do not need to worry about the implementation of the equation (11) and just specify the drift and diffusion at boundaries to get the matrix A from the computer.

[11] For example, in figure 4, $(i,j)|_{E_{i,j,1}} = (i-1,j)$, $(i,j)|_{E_{i,j,2}} = (i,j-1)$, $(i+1,j)|_{E_{i,j,3}} = (i+1,j)$, and $(i,j)|_{E_{i,j,4}} = (i,j+1)$.

2.4 Non-Regular Grids

One advantage of the finite volume method given in equation (11) requires a special attention. One only needs to account for the flows across edges following the discretized equation (11). Hence, the discretization can be applied to more complex grid structures than the regular grid, like the adaptively refined discretization of figure 1c. Now, consider the calculation of the dynamics for one cell given in figure 5: though cell $G_{i,j}$ has non-standard edges, the only difference is that there are more edges to consider, and equation (11) still gives the dynamics across each edge. This flexibility is tested in section 3, where the finite volume method is applied with (i) a uniformly spaced regular grid, (ii) a non-uniformly spaced regular grid, and (iii) a (semi-structured) adaptive grid an shown in figure 1. This is in stark contrast to the finite difference method that does not even generalize to the non-uniform regular grid without losing the mass conservation and positivity.[12] Further, with semi-structured grids, it is unclear how the finite difference method can even be applied. In comparison, even with the semi-structured grid, the finite volume method results in the linear discretized equation given in equation (12). This allows one to outsource the construction of the matrix A of equation (12) to a computer by supplying only the drifts and diffusions at the edges. Hence, even with semi-structured grids defined with adaptive refinements, one can just use matrix operations to compute the dynamics of distributions using equation (12).

2.4.1 Adaptive Refinement

Given the flexibility, the question of finding a reasonable grid still remains. One method of refinement is to iteratively/adaptively add more grid points based on an error metric/estimate from the current solution. Lan et al. (2002) suggests many different refinement metrics, but notes that "normalized gradients" are sufficient for most cases. Following this advice, the adaptation metric tested was based on the normalized gradient where the normalization is by the fraction of the distribution in the cell, i.e.,

$$\text{metric} \approx G_i \cdot \text{drift}\left(\frac{x_i + x_{i+1}}{2}\right). \tag{13}$$

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. Hence, the cell should be refined/split only when both the drift and the fraction of the cell are large.[13] In practice, the normalized drift is a good starting point, but other metrics can be used as well. For example, for the life-cycle model, weighting younger generations leads to faster improvements. This is intuitive as the accuracy of the young affects the accuracy of the old. The finite volume method is flexible enough to handle this intuitive metric of grid refinement.

2.5 One Method to Rule Them All

Though it does not fit with the flow of the paper, one comment is required before the numerical experiments. There is no single method that is the best for all applications. Even *within* the finite volume method, there is a tradeoff between flexibility and speed. Adaptive grids give flexibility, but at the cost of speed (compared to the regular grid). In practice, a regular grid should be used unless a gain from adaptive refinements is expected. For example with the experiments conducted in section 3, using the flexibility leads to a speed loss without any gain for the Ornstein-Uhlenbeck process, but leads to a compression and accuracy gain with economic problems. The Aiyagari-Bewley-Huggett model requires only 58% of grid points to achieve equal accuracy as a regular grid and the lifecycle model 14%. However, this is at the cost of some additional computation. This cost would be regained if one uses the perturbation method introduced in Ahn et al. (2018) since the solution method (without model reduction) scales at $O(N^3)$.[14] Hence, the 14% compression of adaptive method leads the adaptive grid to scale at $0.2\%(= 0.14^3)$ of the regular grid in the perturbation step.

[12] Achdou et al. (2017) generalizes the finite difference to the non-uniform regular grid, but the method introduced there requires adjustments that make the resulting scheme no longer the finite difference method. In fact, one can consider the suggested adjustment as an approximation of the finite volume method, and in the limit of decreasing grid-size, the adjusted scheme converges to the finite volume scheme.

[13] Note that this is why cells should not be adapted only based on the mass, which might seem intuitive initially.

[14] The scale comes from the QZ/Schur-decomposition step. See Ahn et al. (2018) for more details.

#(grid) per dimension	10	20	40	80	160	320	640	1280
Compute Time (sec)	0.0015	0.0024	0.0062	0.030	0.081	0.37	1.97	10.33
Total Variation Metric	5.9e-2	2.3e-2	7.1e-3	2.0e-3	5.4e-4	1.4e-4	3.6e-5	9.2e-6

Table 2: Accuracy and compute time of the finite volume method to the 2-D Ornstein-Uhlenbeck Process with $\theta = (1,1)^T$, $\sigma = (0.1\sqrt{2}, 0.1\sqrt{2})^T$, and $\mu = (0.495, 0.495)^T$.

The tradeoff between methods exists between the Monte Carlo method and the finite volume method as well. Though only the benefits of the finite volume method over the Monte Carlo method are presented as the Monte Carlo method is the literature standard, the two methods scale differently. The Monte Carlo method scales at $O(N^2)$ independent of the dimensionality of the problem, but the finite volume method (or other function approximation methods) scale at $O(N^d)$ where d is the dimensionality of the problem. Therefore, at some scale of the problem the Monte Carlo method will outperform the finite volume method.[15] As tested in section 3, the current generation of economic models have not reached the point where the Monte Carlo methods outperform the function approximation methods. However, with future generations of heterogeneous agent models, the Monte Carlo method can outscale function approximation methods.

The result of a cost-benefit analysis of different methods is application-dependent. As estimating the result of the cost-benefit analysis is hard, one should always test. The companion codes should reduce the implementation time for testing the finite volume method.

3 Numerical Experiment

3.1 Exact Solution: Ornstein-Uhlenbeck Process

First, the finite volume method is tested on a problem with a closed-form solution. Consider the following Fokker-Planck equation

$$\frac{\partial f}{\partial t} = \sum_i \theta \frac{\partial}{\partial x_i} \left[(x_i - \mu) \cdot f \right] + \sum_i \frac{\sigma^2}{2} \frac{\partial^2 f}{\partial x_i^2} \tag{14}$$

where $\theta, \sigma > 0$. This is the Fokker-Planck equation induced by the Ornstein-Uhlenbeck process:

$$\mathrm{d}x_t = \theta(\mu - x_t)\,\mathrm{d}t + \sigma\,\mathrm{d}W_t.$$

One can explicitly check that

$$f(x) = \sqrt{\frac{\theta}{\pi\sigma^2}} e^{-\theta \frac{(x-\mu)^2}{\sigma^2}}$$

is the steady-state distribution of equation (14).[16]

As the exact solution is known, the explicit distance between distributions can be computed instead of differences in some moments. The total variation metric defined as

$$\mu(p,q) = \max_{A \in \mathcal{P}(\Omega)} p(A) - q(A)$$

for two given probability measures p and q is used.[17] The value of the total variation metric from the finite-volume approximation is given in table 2, and shows that the finite volume method is very accurate for computing the steady state distribution of the Ornstein-Uhlenbeck process. As a reference point, the total

[15] There is a different scale problem when one considers the heterogeneous agent model where one only knows individual behaviors on a grid. See appendix A for more details.

[16] When this problem is solved numerically, a finite domain is taken. With the introduction of the boundary, a boundary condition needs to be specified. Usually, the reflecting boundary condition is used. Hence, for the approximation to be equal to the exact solution, the domain needs to be large enough for the given value of σ.

[17] Total variation metric is one of the distance metrics for probability measures. Compared to the Kullback-Leibler divergence or the Wasserstein metric, the total variation distance is easier to compute numerically.

variation distance of two normal distributions, $N(\theta_1, 1)$ and $N(\theta_2, 1)$, is given by $\sqrt{\frac{\pi}{2}}|\theta_1 - \theta_2|$. Therefore, the total variation distance of $9.23 \cdot 10^{-6}$ corresponds to the difference of $7.36 \cdot 10^{-6}$ if reinterpreted as a shift of means of two normal distributions.

The result can be compared to Monte Carlo simulations. For Monte Carlo simulations, agents can be directly simulated from the Ornstein-Uhlenbeck process

$$\mathrm{d}x_t = -\theta(x_t - \mu) \cdot \mathrm{d}t + \sigma \, \mathrm{d}W_t$$

Unfortunately, the computation of the total variation metric is not possible with the Monte Carlo simulation without further approximations,[18] so the relative errors of the first and second moments of the steady state distribution are reported instead. Further, there is no benchmark implementation of the Monte Carlo simulation, and the Monte Carlo simulation can be tuned for specific problems. For this particular simulation, 1) the initial agents are sampled from the exact steady-state distribution, 2) the time discretization of $\Delta t = 0.01$ with 100 time steps are used, and 3) Heun Scheme is used for the discretization of the stochastic process.[19] Starting with the initial draws from the correct distribution is not feasible in practice, but this assumption is made for the numerical comparison as it uniquely pins down the initial distribution and gives the best advantages to the Monte Carlo simulation.[20] Lastly, to approximate sampling errors, the simulation is repeated 100 times to get the distribution of the errors.

Figure 6 and figure 7 show the errors on their computation time of the finite-volume method with regular grid and the Monte Carlo simulations. One can see from the absolute error in the mean in figure 6 that the finite-volume method is multiple orders of magnitude faster than the Monte Carlo simulation. From the same plot, one can see the sampling error of the Monte Carlo simulation, the orange and yellow lines show the 5/95%-quantile values from the Monte Carlo simulations, and the errors are a further two orders of magnitude higher compared to the median value. The error metric on variances leads to similar conclusions. The accuracy of the finite volume method is at the same level as the median value of the 100 Monte Carlo simulations, but faster and scale better than the 5/95% quantile value fo the variances of the 100 Monte Carlo simulation runs.

3.2 Wealth + Income Shock (Aiyagari-Bewley-Huggett) Model

For the first economic application, a heterogeneous agent model with wealth and income heterogeneity, known as the Aiyagari-Bewley-Huggett model in literature, is solved. As this model is standard, the model is not described in detail, see Achdou et al. (2017) for a more detailed description of the model (in continuous time). Households solve the optimization problem

$$\begin{aligned} \max_c & \int_{t=0}^{\infty} e^{-\rho t} u(c) \, \mathrm{d}t \\ & \mathrm{d}a = (r \cdot a + w \cdot z - c) \cdot \mathrm{d}t \\ & \mathrm{d}z = \mu_z(z) \, \mathrm{d}t + \sigma_z(z) \, \mathrm{d}W_t \\ & a \geq \underline{a} \end{aligned}$$

[18] One would compute an approximation of the total variation metric with kernel smoothing, but then the metric is no longer a simple analysis of the accuracy of the Monte Carlo procedure.

[19] The Heun scheme is given by

$$\begin{aligned} x'_{t+\Delta t} &= -\theta(x_t - \mu) \cdot \Delta t + \sqrt{\Delta t} \cdot \sigma \cdot \varepsilon && \varepsilon \sim N(0,1) \\ x_{t+\Delta t} &= -\theta \left(\frac{x_t + x'_{t+\Delta t}}{2} - \mu \right) \Delta t + \sqrt{\Delta t} \cdot \sigma \cdot \varepsilon. \end{aligned}$$

The Heun scheme is considered more accurate compared to the standard Euler-Maruyama scheme given by

$$x_{t+\Delta t} = -\theta \left(x_t - \mu \right) \Delta t + \sqrt{\Delta t} \cdot \sigma \cdot \varepsilon \qquad \varepsilon \sim N(0,1).$$

See Rüemelin (1982) for accuracy comparison between different discretization methods. One can also check Zahri (2009) for the results of other numerical experiments with different Monte Carlo schemes.

[20] This means that one should be careful about accuracy results when reducing the number of simulation steps in the replication file as it is started from the correct distribution.

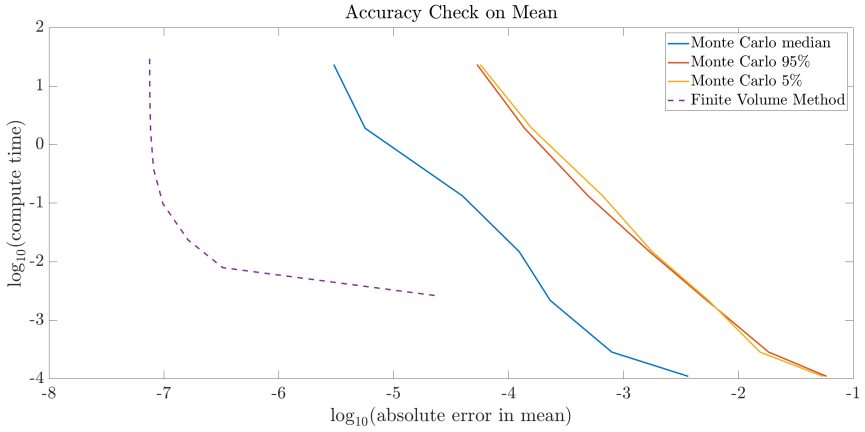


Figure 6: Absolute error in estimates of the mean from the finite volume method and the Monte Carlo simulations. As each Monte Carlo simulation results in different values of the mean, 100 separate simulations were run and the errors from the median, 5% and 95% quantile values are shown.

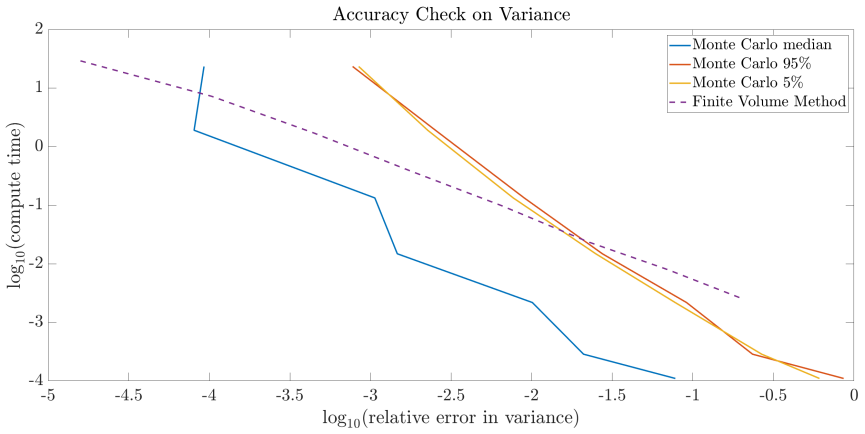


Figure 7: Relative error in estimates of the variance from different methods. As each Monte Carlo simulation results in different values of the variance, 100 separate simulations were run and the errors from the median, 5% and 95% quantile values are shown.

where

a	wealth
z	(idiosyncratic) income state
$\underline{a}$	Borrowing Constraint
r	interest rate on wealth
w	wage
ρ	discount rate
$u(\cdot)$	utility function
W_t	Brownian motion
c	Consumption

Let $c^*(a,z)$ be the optimum value from the optimization problem solved from the dynamic programming, then the dynamics induced for an individual is

$$\mathrm{d}a = (r \cdot a + w \cdot z - c^*(a,z)) \cdot \mathrm{d}t \tag{15}$$
$$\mathrm{d}z = \mu_z(z)\,\mathrm{d}t + \sigma_z(z)\,\mathrm{d}W_t \tag{16}$$
$$a \geq \underline{a}$$

The dynamics above induces the Fokker-Planck equation of

$$\frac{\partial}{\partial t}g(a,z) = -\frac{\partial}{\partial a}\left[(r \cdot a + w \cdot z - c^*(a,z))g(a,z)\right] - \frac{\partial}{\partial z}\left[\mu(z)g(a,z)\right] + \frac{1}{2}\sigma_z(z)^2\frac{\partial^2}{\partial z^2}g(a,z)$$

with the resulting steady-state equation of

$$0 = -\frac{\partial}{\partial a}\left[(r \cdot a + w \cdot z - c^*(a,z))g(a,z)\right] - \frac{\partial}{\partial z}\left[\mu(z)g(a,z)\right] + \frac{1}{2}\sigma_z(z)^2\frac{\partial^2}{\partial z^2}g(a,z) \tag{17}$$

Given the behavior and the aggregate dynamics induced from the behaviors, the economy is closed with equilibrium conditions of

$$K = \int\int a \cdot g(a,z)\,\mathrm{d}a\;\mathrm{d}z$$
$$L = \int\int z \cdot g(a,z)\,\mathrm{d}a\;\mathrm{d}z$$
$$w = (1-\alpha)\cdot\left(\frac{K}{L}\right)^{\alpha}$$
$$r = \alpha\cdot\left(\frac{L}{K}\right)^{1-\alpha}$$

Note that $\widetilde{K} := \dfrac{K}{L}$ determines both w and r. Therefore, $\widetilde{K}$ determines the equilibrium, and in subsequent sections, $\widetilde{K}$ is used as a scalar metric of comparison for different methods. Given the Fokker-Plank equation (17), the discretization can be implemented directly following equation (11) independent of the type of grid under consideration. As noted before, the construction of matrix A of equation (12) can be handled by the computer, and one only needs to specify drifts and diffusions at cell boundaries.

The model is solved for different (uniform) regular grids with adaptive refinements. The equilibrium amount of capital using different grids with the finite volume method is given in figure 8. The finite-volume method without adaptive refinements, shown with red dots, clearly shows convergence as predicted by the theory. The same figure also gives the equilibrium capital resulting from adaptive refinements in the wealth

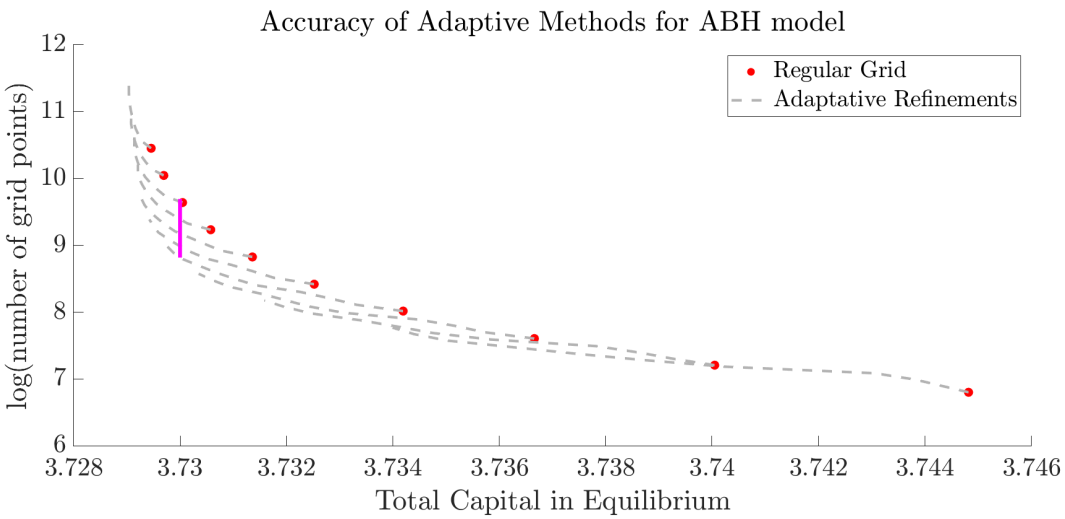


Figure 8: Convergence to the true solution given the number of grid points based on the number of adaptive refinement of the grid. The plot shows multiple regular grids (red dots) and the subsequent iterative refinements (dashed lines). The red dots and dashed lines show clear convergence as proven in theory (x-axis is within 0.1% of the limiting value). At the magenta line, the adaptive methods only require 42% of the grid points of the regular grid to get the same accuracy.

Number of households	10^2	10^3	10^4	10^5	10^6
$\frac{\text{std}(K_t)}{\overline{K}}$	10.31%	2.89%	0.750%	0.275%	0.0948%
$\frac{\max(K_t) - \min(K_t)}{\overline{K}}$	66.63%	19.19%	4.82%	1.72%	0.559%

Table 3: $\overline{K} = 3.7285$ is taken as 3.7285.

dimension based on the following metric.[21]

$$\text{metric} = \underbrace{(r \cdot a_i + w \cdot z_j - c^*(a_i, z_j))}_{\text{drift}} \cdot \underbrace{G_{ij} \cdot a_i}_{\text{normalization}}$$

Hence, in each iteration, a cell is split in half if it is in the top 10% of the metric values or if the neighboring points have been refined further than the cell under consideration.[22] Unlike the Ornstein-Uhlenbeck process in the previous section, adaptive refinements lead to faster convergence (in number of grid points) compared to a non-adapted grid. To get the equilibrium capital value of 3.73 ($\sim$ 0.05% error), one needs 42% of grid points compared to that of the regular grid without adaptive refinements. To reiterate the point made in section 2.5, the speed gain is expected to be 93% ($= 1 - 0.42^3$) for the perturbation method introduced in (Reiter, 2009; Ahn et al., 2018).

This can be compared with the computation of the equilibrium value of capital based on the Monte-Carlo simulation. The sampling error involved in the Monte Carlo simulation is prohibitive enough to make the computation of the equilibrium level of $\widetilde{K}$ impossible in a reasonable time. This can be seen from figure 9 that plots the capital supply given the prices from $\widetilde{K} = 3.7285$ over a simulation period of 5000 years (with time-step of 0.01). Though the capital supply is a constant, the simulated capital supply shows nontrivial

[21] Allow splitting cells in wealth direction, but not in income direction.
[22] These are all hyperparameters to the adaptive refinements and can be adjusted for different applications.

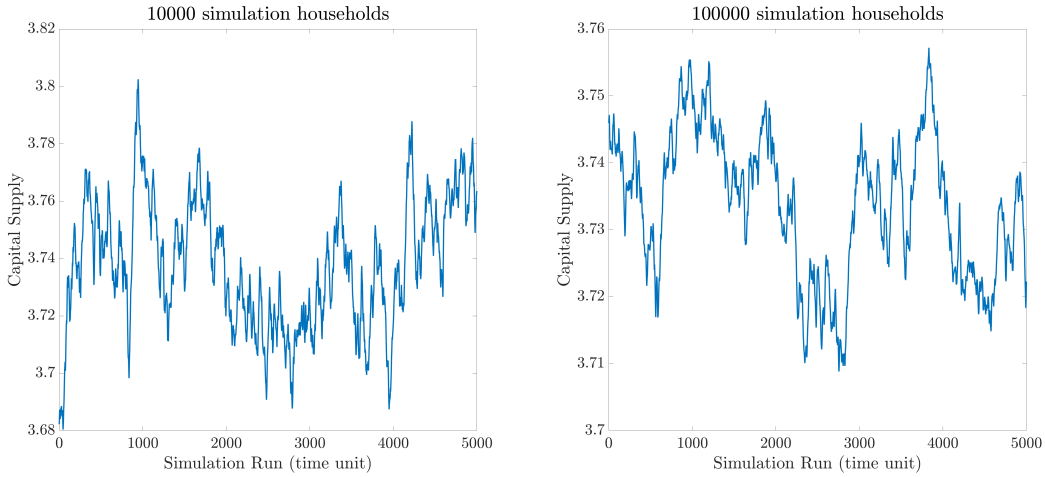


Figure 9: Monte Carlo simulation of capital supply from prices induced from $\widetilde{K} = 3.7285$. As there is no aggregate shock, there is one (constant) value of capital supply. All variations in the figures are from the sampling error. This type of simulations shows what the expected sampling error of the stochastic simulation methods such as Krusell & Smith (1998) given the number of agents.

dynamics.[23] The errors from sample runs are summarized in table 3, where the (normalized) standard deviation and maximum range of the simulated capital are computed over 10,000 years with $\Delta t = 0.01$. Note that even with 100,000 households, the sampling error is 0.275% in standard deviation[24] and 1.72% in the maximum range.[25] Hence, the number of households required to achieve reasonable accuracy can be quite high for the (Wealth+Income)-heterogeneity model. Algan et al. (2014) noted that sampling errors can mask the aggregate dynamics, and this simulation confirms that sampling errors can create dynamics where none exists. This sampling error makes the standard bisection method infeasible for the Monte Carlo simulation within a reasonable time. Hence, one would need to use a different method that would scale worse than $O(\log(n))$ to find the steady state value. In fact, it was not possible to create a robust method that guarantees the convergence to the steady-state $\widetilde{K}$ within a reasonable time.

The computation time of the simulation with 100,000 households is in the scale of hours.[26] This is in contrast to seconds or minutes for the finite volume method that gives the equilibrium value of $\widetilde{K}$ at much higher accuracy than even just the sampling error. This significant speed loss can seem contradictory to the similar scale from the previous experiment on the Ornstein-Uhlenbeck process. However, this arises from the method of approximating behaviors from the dynamic programming. About 80% of the compute time of the Monte Carlo simulation comes from the interpolation. This is because the interpolation locations have to be found repeatedly with Monte-Carlo simulations as the evaluation points change throughout the simulation (see appendix A for a more detailed explanation of this paragraph). In comparison, the interpolation locations need to be found only once for the finite volume method. This means that the Monte Carlo method is expected to scale worse with higher dimensional HJB equations at $\log(n_{\text{HJB}} \cdot n_{\text{simulation}})$. The finite volume method, on the other hand, is expected to scale at $(\log(n_{\text{HJB}}) + n_{\text{simulation}})$. This explains the slowdown of the Monte Carlo method in the Aiyagari-Bewley-Huggett model in comparison to the Ornstein-Uhlenbeck process of the previous section that did not require interpolations. In economics, a method does not exist on its own and the interaction with other parts of the computation need to be considered. The finite volume method (in continuous time) interacts better with the dynamic programming for computing the behaviors of agents than the Monte Carlo simulation leading to improved accuracy with less compute

[23] One can also compute the capital supply by following one agent over time, but a population is simulated as the resulting figure 9 gives a better visualization of the sampling error. The number of agents required in the population would be translated to the length of simulation for the one-agent simulation. Also, sampling of the population mirrors the procedure used under the stochastic simulation algorithms like Krusell & Smith (1998).

[24] L^2-norm

[25] L^∞-norm.

[26] The precise computation time depends on hyperparameters like the burn-in and time-step size.

time.

3.3 Liquid Wealth + Income Shock + Life-cycle Model

For the last numerical experiment, age is introduced to the Aiyagari-Bewley-Huggett model. Under this specification, households solve the optimization problem

$$\begin{aligned}
\max_c \mathbb{E}\left[\int_{t=0}^{\tau} e^{-\rho t}u(c)\,\mathrm{d}t + e^{-\rho\tau}B(a_\tau)\right] \\
\mathrm{d}a &= (r\cdot a + w\cdot\phi_{\text{age}}\cdot z - c)\cdot\mathrm{d}t \\
\mathrm{d}z &= \mu_z(z)\,\mathrm{d}t + \sigma_z(z)\,\mathrm{d}W_t \\
\mathrm{d}\text{age} &= 1 \\
x_{\text{age}} &\in \{\text{alive}, \text{dead}\} \quad \text{poisson process with transition probability } \lambda_{\text{age}} \\
\tau &= \min\{\text{age} : x_{\text{age}} = \text{dead}\} \\
a &\geq \underline{a}
\end{aligned}$$

where the new parameters are

τ	(stochastic) death age
$B(\cdot)$	Bequest motive
ϕ_{age}	Age-Income Gradient

Letting $c^*(a, z, \text{age})$ denote the optimal consumption decision, the induced Fokker-Planck equation is

$$\begin{aligned}
\frac{\partial}{\partial t}g(a,z,\text{age}) = &-\frac{\partial}{\partial a}\left[(r\cdot a + w\cdot z - c^*(a,z,\text{age}))g(a,z,\text{age})\right] \\
&-\frac{\partial}{\partial z}\left[\mu(z)g(a,z,\text{age})\right] + \frac{1}{2}\sigma_z(z)^2\frac{\partial^2}{\partial z^2}g(a,z) \\
&-\frac{\partial}{\partial \text{age}}g(a,z,\text{age}) \\
&-\lambda_{\text{age}}g(a,z,\text{age})
\end{aligned}$$

with the initial condition at age $= 25$ of

$$g(a, z, 25) = g_{25}(a, z)$$

given.

As people in different parts of their life-cycle have different wealth levels, a large gain is expected for this problem from adaptive refinements. Figure 10, where each dot denotes the center of a cell, shows the result of adaptive refinements with the finite volume method. It is clear that the wealth refinements need to differ based on their age, and the finite volume method tracks the distribution well over the life-cycle. This would not be feasible with the finite difference method that requires uniform regular grids (for mass conservation and positivity).

Figure 11 plots the results from uniform regular grids with adaptive refinements on asset grids. For the lifecycle problem, the refinement metric that performed best was

$$\text{metric} = \underbrace{(r\cdot a_i + w\cdot z_j - c^*(a_i, z_j, \text{age}_k))}_{\text{drift}}\cdot\underbrace{G_{ijk}\cdot(150-\text{age})\cdot(a_{i+1}-a_i)}_{\text{normalization}}$$

Hence, in each iteration, a cell was split when it was in the top 100 in the metric values. The notable difference compared to the previous section is the $(150 - \text{age})$ term. This introduces the intuition that computing the distribution of young is more important as the approximations on young affect the values of older households. As can be seen from figure 11, the finite volume method clearly converges (as predicted by theory), and adaptive refinements lead to faster convergence (in number of grid points). For example, as shown by the magenta line in figure 11, one only needs 14.5% of grid points of the regular grid to achieve a similar level of accuracy.

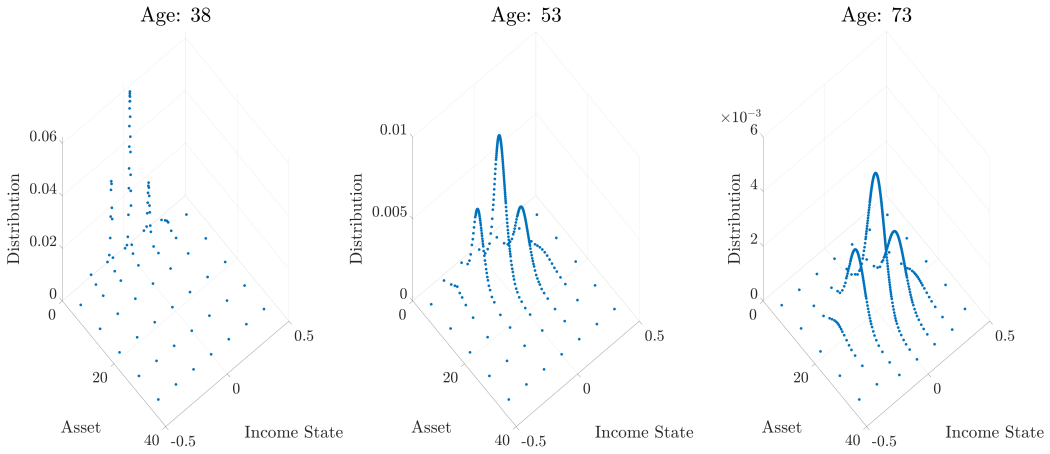


Figure 10: Adaptively refined grid with the finite volume method. Each dot denotes the center of discretization cells.

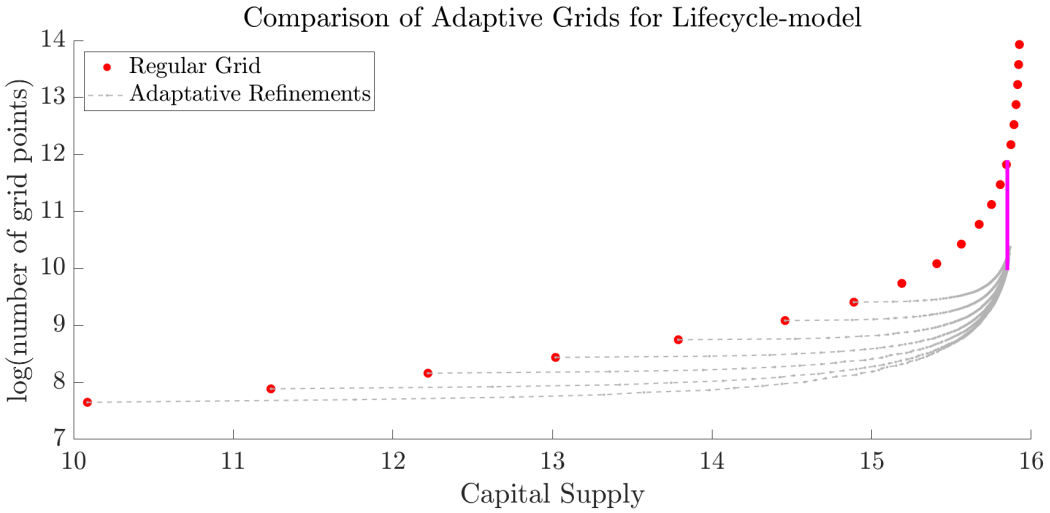


Figure 11: Convergence to the true solution given the number of grid points based on the number of adaptive refinements of the grid for the life-cycle model (partial equilibrium). The plot shows regular grids of different sizes (red dots) with subsequent adaptive refinements (dashed lines). At the magenta line ($K = 15.85$), an adapted grid with 14.5% of grid points is sufficient to match the accuracy of the regular grids.

4 Conclusion

This paper is a bridge paper connecting the mathematics literature on the finite volume method to the economics literature on computing aggregate distributions in heterogeneous agent models. With the flexibility to use non-uniform regular grids and adaptive refinements while retaining the mass conservation and positivity, the finite volume method has benefits over pre-existing methods. For the test problems, the finite volume method outperforms pre-existing methods, and especially the Monte Carlo simulation method.

In applications, the best numerical method will depend very much on the problem at hand, and one should not conclude that the finite volume method is the *best* method for all applications. One should always test different methods, and the intuition introduced in this paper with the companion codes should allow researchers to test the finite volume method without incurring too many programming hours. The open-sourced code, however, is not at the limit of the applicability of the finite-volume method. For future research, one can consider using even more non-structured grids for the finite-volume method presented in Eymard et al. (2000) and using parallel computing.

References

Achdou, Y., Han, J., Lasry, J.-M., Lions, P.-L., & Moll, B. (2017, August). *Income and wealth distribution in macroeconomics: A continuous-time approach* (Working Paper No. 23732). National Bureau of Economic Research. Retrieved from `http://www.nber.org/papers/w23732` doi: 10.3386/w23732

Ahn, S., Kaplan, G., Moll, B., Winberry, T., & Wolf, C. (2018). When inequality matters for macro and macro matters for inequality. *NBER macroeconomics annual*, *32*(1), 1–75.

Algan, Y., Allais, O., & Den Haan, W. J. (2008). Solving heterogeneous-agent models with parameterized cross-sectional distributions. *Journal of Economic Dynamics and Control*, *32*(3), 875–908.

Algan, Y., Allais, O., Den Haan, W. J., & Rendahl, P. (2014). Solving and simulating models with heterogeneous agents and aggregate uncertainty. In *Handbook of computational economics* (Vol. 3, pp. 277–324). Elsevier.

Axelsson, O., & Gustafsson, I. (1979). A modified upwind scheme for convective transport equations and the use of a conjugate gradient method for the solution of non-symmetric systems of equations. *IMA Journal of Applied Mathematics*, *23*(3), 321–337.

Brumm, J., & Scheidegger, S. (2017). Using adaptive sparse grids to solve high-dimensional dynamic models. *Econometrica*, *85*(5), 1575–1612.

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.

Eymard, R., Gallouët, T., & Herbin, R. (2000). Finite volume methods. *Handbook of numerical analysis*, *7*, 713–1018.

Ferm, L., Lötstedt, P., & Sjöberg, P. (2006). Conservative solution of the fokker–planck equation for stochastic chemical reactions. *BIT Numerical Mathematics*, *46*(1), 61–83.

George, D. (2011). Adaptive finite volume methods with well-balanced riemann solvers for modeling floods in rugged terrain: Application to the malpasset dam-break flood (france, 1959). *International Journal for Numerical Methods in Fluids*, *66*(8), 1000–1018.

George, D. L., & LeVeque, R. J. (2006). Finite volume methods and adaptive refinement for global tsunami propagation and local inundation. *Science of Tsunami Hazards*, *24*(5), 319–328.

Krusell, P., & Smith, A. A., Jr. (1998). Income and wealth heterogeneity in the macroeconomy. *Journal of political Economy*, *106*(5), 867–896.

Lan, C. W., Liu, C. C., & Hsu, C. M. (2002). An adaptive finite volume method for incompressible heat flow problems in solidification. *Journal of Computational Physics*, *178*(2), 464497.

Li, Y., Liu, J.-L., Chao, T.-S., & Sze, S. (2001). A new parallel adaptive finite volume method for the numerical simulation of semiconductor devices. *Computer physics communications*, *142*(1-3), 285–289.

Reiter, M. (2009). Solving heterogeneous-agent models by projection and perturbation. *Journal of Economic Dynamics and Control*, *33*(3), 649–665.

Ríos-Rull, J.-V. (1999). Computation of equilibria in. *Computational methods for the study of dynamic economies*, 238.

Rüemelin, W. (1982). Numerical treatment of stochastic differential equations. *SIAM Journal on Numerical Analysis*, *19*(3), 604–613.

Winberry, T. (2018). A method for solving and estimating heterogeneous agent macro models. *Quantitative Economics*, *9*(3), 1123–1151.

Young, E. R. (2010). 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.

Zahri, M. (2009). Stochastic diffusion problems: Comparison between euler-maruyama and runge-kutta schemes. *Journal of Numerical Mathematics and Stochastics*, *1*(1), 65–76.

A A Practical Aside: Interpolations

Interpolations are ubiquitous in the methodology literature. In fact, the procedures introduced under Ríos-Rull (1999) and Young (2010) rely on interpolations. Though people write "... then we just interpolate..." frequently without giving much attention, interpolations can be the most costly bottleneck. As an extreme example, Brumm & Scheidegger (2017) reports that 99% of their calculation time is spent on interpolations, and in the Monte Carlo simulation of the Aiyagari-Bewley-Huggett model, 80% of calculation time was spent on interpolations. Sometimes, this is just an intrinsic limit of the given problem, but it can be a result of an inefficient implementation of the interpolation. To see why, consider the linear interpolation in one-dimension. The interpolated value at x is given by

$$\widetilde{f}(x) = f(x_i)\frac{x_{i+1} - x}{x_{i+1} - x_i} + f(x_{i+1})\frac{x - x_i}{x_{i+1} - x_i} \qquad x \in [x_i, x_{i+1}] \tag{18}$$

where $\{x_i\}$'s are the knot points of the interpolation. The calculation of equation (18) is actually two separate operations. One searches for the index i such that $x \in [x_i, x_{i+1}]$ first, and then evaluates the algebraic expressions given the $x_i, x_{i+1}, f(x_i)$ and $f(x_{i+1})$. Costs of the two operations scale differently. For the search problem, if one just checks through all x_i's to find the interpolation location, then the operation scales at $O(n_{\text{knots}})$ as one reads n_{knots} points. In applications with structured knot points, one can do better and the search problem can be reduced to scale at $O(\log(n_{\text{knots}}))$, for example, by using the bisection method. On the other hand, the evaluation of the interpolated value scales at $O(1)$.[27] Even though log scale is fast, it still scales worse than the constant scale, and even small value of $\log(n_{knots})$ lead to a speed loss of a few hundred percentages as it is multiplicative. Therefore, the *search* for the interpolation locations is the expensive part of the interpolation, not the *evaluation*. This applies to all "local" interpolation methods like the linear or the cubic interpolation.

Now, if the interpolation is repeated N times and the interpolation locations are found repeatedly, the computation scales at

$$O(n_{\text{evaluation}} \cdot \log(n_{\text{knots}}) \cdot N) \tag{19}$$

[27] It is 9 in equation (18).

However, if one can reuse the search results, one can reduce the scale to

$$O(n_{\text{evaluation}} \cdot \log(n_{\text{knots}}) + n_{\text{evaluation}} \cdot N) \tag{20}$$

This difference in the scale makes a very practical difference for large N, and some numerical methods can reuse the search results more naturally than others.

With the finite volume method, the interpolation locations depend only on the grid, not on the parameter values. Therefore, the search problem only needs to be handled once. Unfortunately, with the Monte Carlo simulation, the interpolation location changes every evaluation, not just with new parameters. Hence, the scale of the Monte Carlo simulation scales as equation (19) while the finite volume method scales as equation (20). For many heterogeneous agent models, n_{knots} is the number of grid points required in approximating the value function, and exhibits the curse of dimensionality. The Monte Carlo simulation will subsequently inherit and amplify the curse of dimensionality from the dynamic programming. In comparison, the finite volume method only needs to incur the cost once.

This interpolation difference exists even within the function approximation framework, and explains some of the speed differences between continuous time and discrete time methods. In continuous time, the search problem only depends on the grids, but not on the specific parameter values. However, under discrete time methods, the search problem depends on both the grids *and* parameter values, and the search problem needs to be resolved for new parameter values. Of course, in practice, the interpolation locations will not change too much for a new set of parameters, and an implementation can and should leverage previous search results to reduce the scale dependence from $\log(n_{\text{knots}})$. However, this is problem-dependent and requires programming hours for customization. Hence, in practice, many practitioners just rebuild the interpolation locations from scratch in every iteration resulting in the effective scale of $O(n_{\text{evaluation}} \cdot \log(n_{\text{knots}}) \cdot N)$ though the intrinsic scale of the problem might be smaller.[28] Of course, this recycling of search results is automatic in continuous time.

B Implementation

For numerical experiments, two different implementations in MATLAB are used. One data structure is for uniform/non-uniform regular grids. The other data structure allows the introduction of any rectangular-shaped cells, such as those seen in figure 1c. One pays for the added flexibility with the computation cost. As the open-sourced codes are provided at `<https://github.com/sehyoun/adaptive_finite_volume>` and the specific implementation can be updated in the future, the technical details of implementations are delegated to the codes and the documentation is available at `<https://sehyoun.com/adaptive_finite_volume>`.

[28] I have not yet seen any example of an economist recycling search results for a efficiency gain in discrete time. This is sensible if one takes into account the programming time, but this would restrict the size of problems researchers would handle.