

# Heterogeneous-Agents Macroeconomics in Continuous Time<sup>1</sup>

Lecture Notes<sup>2</sup>

Riccardo A. Cioffi

This version: January 14, 2026

Latest version at: <https://github.com/ric-cioffi/hact-notes>

---

<sup>1</sup>These lecture notes were developed as part of the teaching material for a PhD class taught at the Paris School of Economics. The material *heavily* borrows from a class taught by Benjamin Moll while I was a student at Princeton, as well as from other classes taught there, most notably by Gianluca Violante and Jonathan Payne. I also received a lot of help from one of my students, Leonardo Endrizzi, as well as other PSE students who provided very useful feedback. While these notes would not have been possible without them, it goes without saying that all errors are exclusively mine. If you find error, typos, or would simply like to discuss the material in these notes please write me an email at [riccardo.cioffi@psemail.eu](mailto:riccardo.cioffi@psemail.eu)

<sup>2</sup>These notes are licensed under a CC BY-NC 4.0 International License. They are free to copy, distribute, and adapt for non-commercial purposes. Commercial use is not permitted.



# Contents

<b>1 Deterministic Optimal Control</b>	<b>1</b>
1.1 Hamiltonian . . . . .	1
1.2 The Neoclassical Growth Model . . . . .	3
1.2.1 Phase Diagram . . . . .	5
1.2.2 Numerical Solution . . . . .	8
<b>2 Deterministic HJB Equations</b>	<b>11</b>
2.1 The Hamilton-Jacobi-Bellman Equation . . . . .	11
2.1.1 An Alternative Derivation . . . . .	13
2.2 Numerical Solution . . . . .	14
2.2.1 Finite-differences . . . . .	15
2.2.2 Upwind Scheme . . . . .	17
2.2.3 Discretized HJB Equation . . . . .	20
2.2.4 Explicit and Implicit Methods . . . . .	22
2.2.5 Results . . . . .	26
2.3 Some Practical Considerations . . . . .	28
<b>3 Stochastic Calculus</b>	<b>31</b>
3.1 Stochastic Processes . . . . .	32
3.1.1 Continuous Processes . . . . .	32
3.1.2 Discrete Processes . . . . .	38
3.1.3 Continuous-time Markov Chains . . . . .	40
3.2 Stochastic Calculus . . . . .	42
3.2.1 Differential Algebra . . . . .	43
3.2.2 Itô's Lemma . . . . .	44
3.3 The Kolmogorov Forward Equation . . . . .	47
3.4 Infinitesimal Generators . . . . .	48
3.4.1 Generator of a CTMC . . . . .	49
3.4.2 Generators of General Processes . . . . .	52
3.5 Example: Ornstein-Uhlenbeck process . . . . .	53
3.5.1 Numerical Solution . . . . .	54

<b>4 Stochastic HJB Equations</b>	<b>59</b>
4.1 Stochastic Optimal Control . . . . .	59
4.1.1 Example: The RBC Model . . . . .	62
4.2 Boundary Conditions . . . . .	63
4.2.1 Probability Flux and Conservation . . . . .	64
4.2.2 Types of Boundary Conditions . . . . .	65
4.2.3 Practical Implementation . . . . .	67
4.3 Aiyagari in Continuous-Time . . . . .	69
4.3.1 Setup . . . . .	69
4.3.2 HJB and KF Equations . . . . .	69
4.3.3 Equilibrium . . . . .	71
4.4 Numerical Solution . . . . .	72
4.4.1 Upwind Scheme . . . . .	72
4.4.2 Matrix Representation . . . . .	73
4.4.3 Algorithm . . . . .	75
4.5 Results . . . . .	77
<b>5 Aggregate Shocks</b>	<b>79</b>
5.1 Introduction . . . . .	79
5.1.1 A Brief History of Macro . . . . .	79
5.1.2 Aggregation . . . . .	82
5.2 Aggregate Shocks in Continuous Time . . . . .	86
5.2.1 General Framework . . . . .	86
5.2.2 The Problem with Rational Expectations . . . . .	87
5.2.3 Krusell and Smith . . . . .	88
5.2.4 Near-Aggregation . . . . .	96
<b>Appendices</b>	<b>101</b>
A Deterministic Optimal Control . . . . .	101
A.1 How to draw a phase diagram . . . . .	101
B Deterministic HJB Equations . . . . .	102
B.1 The Envelope Theorem . . . . .	102
B.2 Non-uniform Grids . . . . .	103
B.3 Non-Convexities . . . . .	104
B.4 Code for the Neoclassical Growth Model . . . . .	107
C Stochastic Calculus . . . . .	109
C.1 KFE Derivation . . . . .	109
C.2 $p$ -Variation . . . . .	112
D Stochastic HJB Equations . . . . .	114
D.1 Code for the Aiyagari model . . . . .	114

# Chapter 1

## Deterministic Optimal Control

This lecture serves mostly as an introduction to continuous time modeling.<sup>1</sup> Students who have already worked in continuous time can likely skip it, with the only possible exception being section 1.2.2 where we go over the numerically approximation of a system of linear ordinary differential equations using a finite-difference method.

Also, we hereby use the neoclassical growth model as a running example, but our analysis of the model itself is quite minimalistic. If you're not familiar with this specific model and would like to know more, Acemoglu (2008, Chapter 8) has a more in-depth treatment.

### 1.1. Hamiltonian

In general, a deterministic continuous-time optimal control problems can be written as:

$$\begin{aligned} v(x_0) &= \max_{\alpha(t)} \int_0^{\infty} e^{-\rho t} r(x(t), \alpha(t)) dt \\ \text{s.t. } &\dot{x}(t) = f(x(t), \alpha(t)) \end{aligned}$$

for  $t \geq 0$  and  $x(0) = x_0$  given.<sup>2</sup>

The value function  $v(x_0)$  is the maximum with respect to a control vector,  $\alpha(t) \in A \subseteq \mathbb{R}^k$ , of an instantaneous return function,  $r : X \times A \rightarrow \mathbb{R}$ , which depends on the state vector,  $x(t) \in X \subseteq \mathbb{R}^m$ , and the control vector, discounted at a constant rate  $\rho \geq 0$ .

The state vector  $x$  is assumed to evolve according to some differential equation  $\dot{x}(t) = f(x(t), \alpha(t))$  for  $t \geq 0$ , where  $f : X \times A \rightarrow \mathbb{R}^m$  is called the law of motion for the state. Then, to close the problem, we also usually need an initial (or terminal) condition on the state. In this specification  $x(0) = x_0$  is the initial condition.

---

<sup>1</sup>Due to the nature of this document being lecture notes, chapters will sometimes be equivalently referred to as "lectures".

<sup>2</sup>Dotted variable are generally used to indicate the derivative with respect to time, with one dot indicating the first derivative  $\dot{x}(t) \equiv \frac{\partial x(t)}{\partial t}$ , two dots indicating the second derivative etc.

In order to solve this problem, we can write down the so-called **current value Hamiltonian**  $\mathcal{H}$ :

$$\mathcal{H}(x(t), \alpha(t), \lambda(t)) = r(x(t), \alpha(t)) + \lambda(t)f(x(t), \alpha(t))$$

which is given by the sum of the instantaneous return function,  $r(x(t), \alpha(t))$ , and the product of the law of motion for the state,  $f(x(t), \alpha(t))$ , and a so-called co-state vector  $\lambda(t) \in \mathbb{R}^m$ .<sup>3</sup>

From this, we can obtain a set of conditions that are both necessary (under some regularity conditions) and sufficient (under regularity and concavity conditions) for optimality. These conditions are:

$$\mathcal{H}_\alpha(x(t), \alpha(t), \lambda(t)) = 0 \quad (1.1)$$

$$\mathcal{H}_x(x(t), \alpha(t), \lambda(t)) = -\dot{\lambda}(t) + \rho\lambda(t) \quad (1.2)$$

$$\mathcal{H}_\lambda(x(t), \alpha(t), \lambda(t)) = \dot{x}(t) \quad (1.3)$$

that is,

- ▶ Equation (1.1) requires that the derivative of the Hamiltonian  $\mathcal{H}$  with respect to the control vector  $\alpha(t)$  equals 0, and is the equivalent of the standard FOC for maximization.
- ▶ Equation (1.2) requires that the derivative of  $\mathcal{H}$  with respect to the state vector  $x(t)$  equals minus the time derivative of the co-state vector plus the discount rate times the co-state vector.<sup>4</sup>
- ▶ Equation (1.3) requires that the derivative of  $\mathcal{H}$  with respect to the co-state vector  $\lambda(t)$  equals the law of motion for the state  $\dot{x}(t)$  and it simply boils down to rewriting the constraint  $\dot{x} = f(x, \alpha)$

This gives us a system of three differential equations in three unknowns which, together with boundary conditions for the state and co-state variables, will determine the solution to our problem. When it comes to the boundary conditions, we already have an initial condition for the state variable  $x(0) = x_0$  which we now pair with a transversality condition for the co-state variable  $\lambda(t)$ :

$$\lim_{t \rightarrow \infty} e^{-\rho t} \lambda(t)x(t) = 0 \quad (1.4)$$

which requires that the discounted product between state and co-state variables equals zero. In other words, the transversality condition requires that neither the state nor the co-state variable “explodes” to infinity.<sup>5</sup>

---

<sup>3</sup>Just as the multiplier in standard Lagrangean maximization, the co-state vector  $\lambda(t)$  is simply an auxiliary variable that is useful to solve the problem, though as we will see it will often also have a natural economic interpretation. Notice also that the co-state vector  $\lambda(t)$  has the same dimensionality of the state vector  $x(t)$ .

<sup>4</sup>In other disciplines (especially physics) eq. (1.2) is often written as  $\mathcal{H}_x(\cdot) = -\dot{\lambda}(t)$ . The difference is due to the way that we defined the Hamiltonian and the fact that we are expressing everything in “current-value” terms. Naturally, the solution is exactly identical.

<sup>5</sup>Where by “explode” we mean that their product cannot grow at a rate bigger than  $\rho$ .

It is also important to note that, while we have an *initial* condition for the state  $x(t)$ , we effectively only have a *terminal* condition for the co-state  $\lambda(t)$  in eq. (1.4). This means that the value of the co-state at time  $t = 0$ ,  $\lambda(0)$ , is actually free (i.e. not predetermined). As we will see, this is particularly important because it is what will allow us to get a (stable) solution to our problem.

## 1.2. The Neoclassical Growth Model

To make our approach operational, we can apply it to the standard Neoclassical Growth Model.<sup>6</sup> The model consists of a representative household with utility function:

$$U = \int_0^\infty e^{-\rho t} u(c(t)) dt$$

where  $u(c(t))$  is the instantaneous utility function of (per-capita) consumption  $c(t) \geq 0$  and  $\rho \geq 0$  is the discount rate (as opposed to the discount factor  $\beta$  in the corresponding discrete-time model).<sup>7</sup>

The technology in the economy is:

$$y(t) = F(k(t)) \quad (1.5)$$

with  $y(t)$  being output,  $k(t) \geq 0$  the capital stock, and  $F(k(t))$  the production function – which is assumed to be increasing,  $F'(k) > 0$ , and concave,  $F''(k) < 0$ . The law of motion for the capital stock is given by:

$$\dot{k}(t) = i(t) - \delta k(t) \quad (1.6)$$

where  $i(t)$  represents investment and  $\delta$  is the capital depreciation rate. The household's budget constraint is given by:

$$c(t) + i(t) = y(t) \quad (1.7)$$

Finally, to close the model we assume that the economy start at period  $t = 0$  with an initial capital stock  $k(0) = k_0$ .

We can then write the problem in sequential form as:

$$\begin{aligned} v(k_0) &= \max_{c(t)} \int_0^\infty e^{-\rho t} u(c(t)) dt \\ &\text{s.t.} \\ &\quad \dot{k}(t) = F(k(t)) - \delta k(t) - c(t) \\ &\quad k(0) = k_0 \end{aligned}$$

where, in terms of the notation introduced before, we have:

---

<sup>6</sup>For a more detailed explanation of the model, see Acemoglu 2008, Chapter 8.

<sup>7</sup>If you are new to continuous-time models, recall that the discount factor and the discount rate are inversely related: a higher discount factor means that the household is more patient, where instead a higher discount rate instead means that the household is *less* patient.

- ▶ Consumption  $c(t)$  is the control vector  $\alpha(t)$ .
- ▶ The utility function  $u(c(t))$  is the return function  $r(x(t), \alpha(t))$ .
- ▶ The capital stock  $k(t)$  is our state vector  $x(t)$ , which also means the law of motion for capital gives us the law of motion for the state:

$$f(x(t), \alpha(t)) = F(k(t)) - \delta k(t) - c(t)$$

In order to simplify notation, from now on whenever not strictly necessary we will avoid writing the time dependence of variables; it is implied that all variables are a function of time so for example when writing  $k$  we really mean  $k(t)$ .

We now have all the elements to apply the Hamiltonian approach of the previous section. Just as before, we start by writing the present-value Hamiltonian  $\mathcal{H}$  as the return function plus the law of motion for the state multiplied by the co-state  $\lambda$ :

$$\mathcal{H}(k, c, \lambda) = u(c) + \lambda [F(k) - \delta k - c].$$

The derivatives with respect to  $c$ ,  $k$  and  $\lambda$  are given by:

$$\begin{aligned}\mathcal{H}_c(k, c, \lambda) &= u'(c) - \lambda \\ \mathcal{H}_k(k, c, \lambda) &= \lambda [F'(k) - \delta] \\ \mathcal{H}_\lambda(k, c, \lambda) &= F(k) - \delta k - c\end{aligned}$$

and, after imposing the necessary and sufficient conditions for optimality (and rearranging) we get:

$$\begin{aligned}\lambda &= u'(c) \\ \dot{k} &= F(k) - \delta k - c \\ \dot{\lambda} &= \rho\lambda - \lambda [F'(k) - \delta]\end{aligned}$$

where, just to reiterate, the last two equations essentially give us back the laws of motion for the state and co-state variables, respectively.

To solve the model, we then have to combine the necessary and sufficient conditions for optimality with the initial condition for the state  $k(0) = k_0$  and the transversality condition for the co-state  $\lim_{t \rightarrow \infty} e^{-\rho t} \lambda(t) k(t) = 0$ . Together with the FOCs, we therefore have a system of first-order ordinary differential equations (ODE) that can be easily solved to find the optimal consumption and capital stock paths.

In order to gain some intuition about our problem, the first thing that we will do is to draw a so-called **phase diagram** of the system, which is possible because we are in the one dimensional case (the state variable  $x(t)$  is a scalar). The phase diagram is particularly useful because it describes the evolution of the system, which will deliver us some insights on the dynamics of the model (and on its solution). Generally, phase diagrams are often drawn in the (state, co-state) space. However, in order to have a more direct interpretation,

in this specific example it will be useful to draw it in the  $(c, k)$  space instead.<sup>8</sup>

Let's therefore start by rearranging the system of ODEs to have them in terms of  $c$  and  $k$  (as opposed to  $\lambda$  and  $k$ ). From the FOC  $\lambda = u'(c)$ , using the chain rule, we can obtain the derivative of  $\lambda$  with respect to  $t$  as:

$$\dot{\lambda}(t) = u''(c(t))\dot{c}(t)$$

and we can rewrite the law of motion for the co-state variable as:

$$u''(c)\dot{c} = u'(c)(\rho + \delta - F'(k)) \quad (1.8)$$

which the alert reader might recognize as nothing else than the consumption Euler equation and which is, with the appropriate adjustments, exactly the same as it would be in the discrete time formulation of the model. Similarly, we can also rewrite the transversality condition as:

$$\lim_{t \rightarrow \infty} e^{-\rho t} u'(c(t))k(t) = 0 \quad (1.9)$$

Finally, to further simplify the system, recall that the coefficient of relative risk aversion is simply given by:

$$\gamma(c) = -\frac{u''(c)c}{u'(c)}$$

which means we can rewrite the Euler equation (1.8) as:

$$\frac{\dot{c}}{c} = \frac{1}{\gamma(c)}(F'(k) - \rho - \delta).$$

and the system that we are going to solve is therefore given by:

$$\frac{\dot{c}}{c} = \frac{1}{\gamma(c)}(F'(k) - \rho - \delta) \quad (1.10)$$

$$\dot{k} = F(k) - \delta k - c \quad (1.11)$$

together with the initial condition  $k(0) = k_0$  and the transversality condition (1.9).

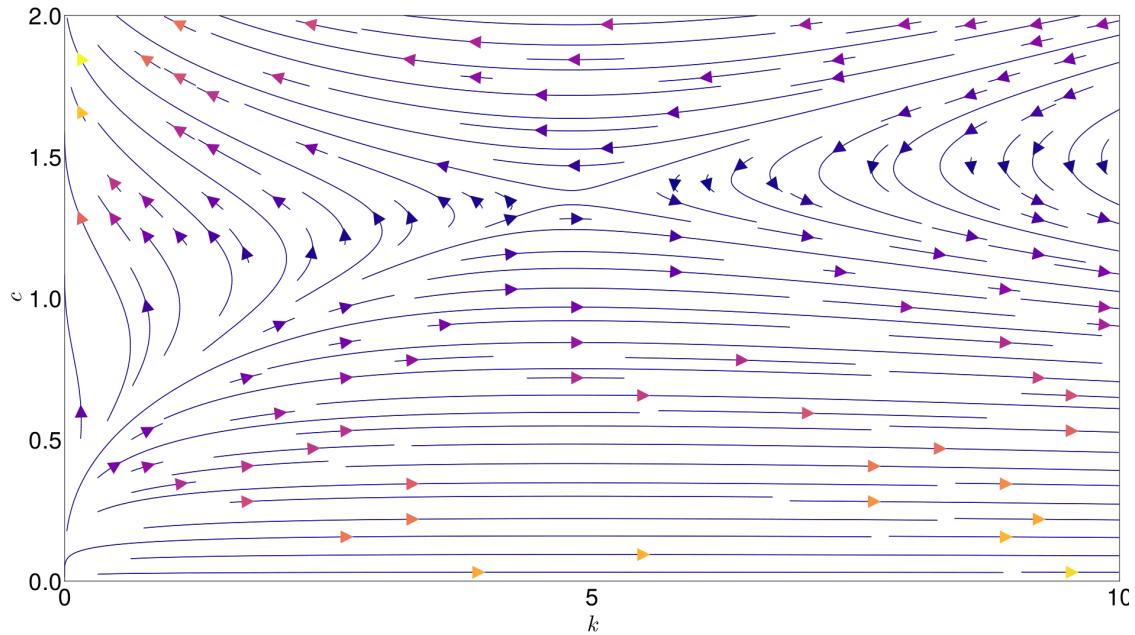
### 1.2.1. Phase Diagram

Figure 1.1 plots the phase diagram in the consumption-capital space. It represents how any point in the  $(c, k)$ -space evolves over time according to our system given by eqs. (1.10) and (1.11). The colors of the arrows indicate the speed at which each point moves in the state space (i.e. essentially the values of  $\dot{c}$  and  $\dot{k}$ ): darker colors (blue and violet) represent lower speeds, and lighter colors (red and yellow) higher speeds. So, by taking any point in the phase diagram as an initial condition and following the arrows, the phase diagram tells us how consumption and capital will evolve over time.<sup>9</sup>

---

<sup>8</sup>If you are not familiar with phase diagrams, section A.1 describes how to draw one.

<sup>9</sup>In fact, by simply looking at fig. 1.1 we can already see that there is something going on in the middle of the graph (at about  $k \approx 5$  and  $c \approx 1.4$ ) as there seems to be a sort of basin where the direction of the



**Figure 1.1:** Phase diagram

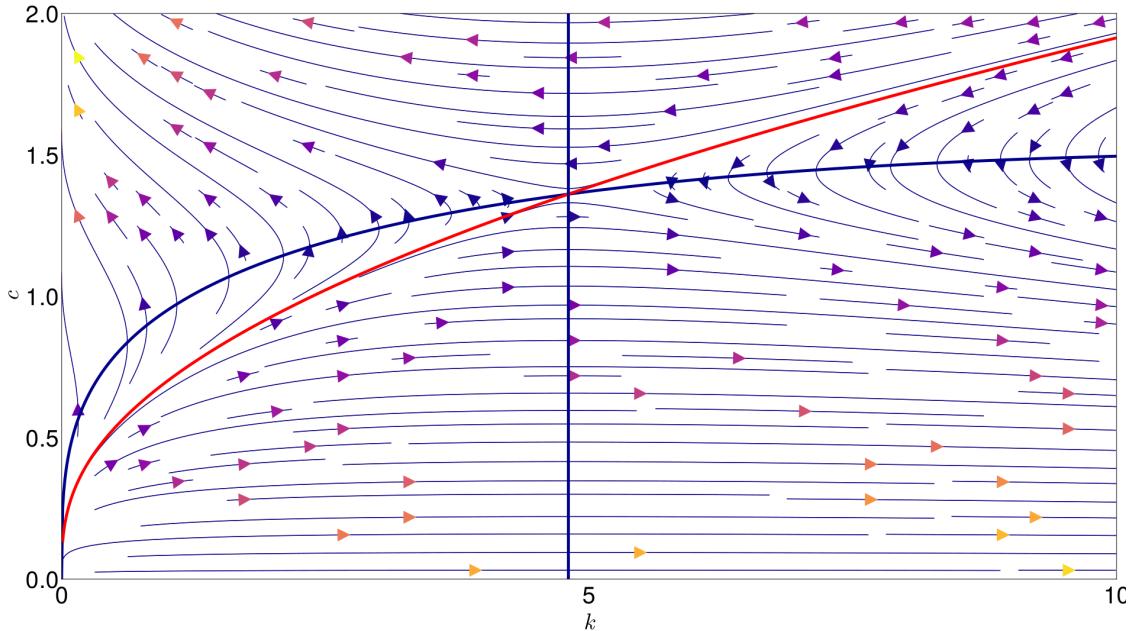
Before looking at the actual solution of the system, it is instructive to better understand what the phase diagram tells us:

1. If the system were to start from a sufficiently low level of initial consumption  $c(0)$  (for any level of capital  $k(0)$ ), the phase diagram tells us that the consumption-capital pair will start going to the right and eventually consumption will start to decrease while capital continues to increase.<sup>10</sup> In fact, it can be shown that, starting from such initial condition, consumption will eventually reach 0 in finite time,  $c(T) = 0$ , while capital will converge to a finite level,  $k(T) = k_{\max}$ . However, this cannot be a solution to our system because it would violate the transversality condition eq. (1.9). So any pair of points  $(c_0, k_0)$  in which initial consumption is “too low” (recall that  $k(0) = k_0$  is given) cannot be a solution of our problem because it would eventually violate the transversality condition; what we mean by “too low” will be evident shortly.
2. If instead the system were to start from a sufficiently high level of consumption  $c(0)$ , the phase diagram tells us that the consumption-capital pair will start going up and eventually capital will start to decrease while consumption continues to increase. In fact, it can be shown that, starting from such initial condition, capital will eventually reach 0 while consumption remains positive. However, this is not possible because it would violate the feasibility condition (zero capital implies zero production which, by the budget constraint, requires zero consumption). So any pair of points  $(c_0, k_0)$  in

---

arrows change. As we will see shortly, this is indeed the steady state of our system.

<sup>10</sup>While it is a bit hard to see from fig. 1.1, all the arrows in the south-east quadrant of the figure are actually slightly pointing downward.



**Figure 1.2:** Phase diagram with saddle path

which initial consumption is “too high” cannot be a solution of our problem because it would eventually violate the feasibility condition.

Hence, our phase diagram tells us that initial consumption has to be neither too low nor too high, or the system will eventually violate either the transversality or the feasibility condition, respectively. Nonetheless, as we will see very soon, the system will in fact converge to the (unique) solution of our system provided that initial consumption is “just right”. Figure 1.2 plots in red all values of initial consumption (for each level of initial capital) that converge to the solution of our system. The set of all the initial pairs  $(c_0, k_0)$  that eventually converge to the solution is also called the equilibrium path or saddle path. Before seeing how we can find such equilibrium path, it is useful to first find the steady-state to our system, i.e. the point of convergence for all consumption-capital pairs that lie on the equilibrium path.

By definition the steady state of our system will be such that the time derivatives of both consumption and capital are equal to zero,  $\dot{k} = \dot{c} = 0$ . Hence, from eqs. (1.10) and (1.11) we immediately get:<sup>11</sup>

$$\dot{c} = 0 \implies F'(k^*) = \rho + \delta \quad (1.12)$$

$$\dot{k} = 0 \implies c^* = F(k^*) - \delta k^*. \quad (1.13)$$

Figure 1.2 plots the solution to these two equations in blue: The vertical line is the solution to eq. (1.12) and represents the locus of points where  $\dot{c} = 0$ , while the concave

<sup>11</sup>Notice that this is exactly the same solution of the corresponding model in discrete time just with  $\rho = 1/\beta - 1$ .

blue line is eq. (1.13) and represents the locus of points where  $\dot{k} = 0$ .<sup>12</sup> Obviously enough, the intersection of these two lines is therefore the steady-state of our system.

If we assume a functional form for the production function, e.g.  $F(k(t)) = Ak(t)^\alpha$ , we can then find the steady state in closed form:

$$k^* = \left( \frac{\alpha A}{\rho + \delta} \right)^{\frac{1}{1-\alpha}}$$

$$c^* = A \left( \frac{\alpha A}{\rho + \delta} \right)^{\frac{\alpha}{1-\alpha}} - \delta \left( \frac{\alpha A}{\rho + \delta} \right)^{\frac{1}{1-\alpha}}$$

Given our discussion about the initial value of consumption lying above or below the saddle path, one might believe that unless initial consumption happens to lie exactly on the equilibrium path, the system will diverge. This is an incomplete, if not incorrect, characterization of the solution to the system. In fact, it is crucial to point out that the system will *always* converge to steady-state equilibrium and it is not a “knife-edge” scenario.<sup>13</sup> The reason is that, as we mentioned before, the co-state variable  $\lambda$  (and therefore consumption) is *free* at time 0. Hence,  $\lambda_0$  can always freely adjust to lie exactly on the saddle path. That is to say, at time zero, for any given initial level of capital  $k(0) = k_0$ , consumption will “jump” to lie exactly on the saddle path. Then  $(k(t), c(t))$  will travel monotonically towards the unique steady state of the system. We have therefore shown (rather informally) that there exists a unique equilibrium path, that the system converges monotonically to the unique steady state and that it does so starting from any initial capital stock  $k_0$ .

### 1.2.2. Numerical Solution

To conclude this lecture, let’s see how to find the saddle path of the Neoclassical Growth Model using a numerical scheme.<sup>14</sup> Recall that the system of ODEs is given by:

$$\dot{k} = F(k) - \delta k - c$$

$$\frac{\dot{c}}{c} = \frac{1}{\gamma(c)} (F'(k) - \delta - \rho).$$

The most natural way to solve this system numerically is to simply find a way to approximate the time derivatives and to solve the system of equations forward in time starting from some initial condition (while figuring out a way to ensure the system converges to

<sup>12</sup>Notice in fact that the arrows crossing the  $\dot{c} = 0$  line are always parallel to the  $k$  axis at that point and, similarly, the arrows crossing the  $\dot{k} = 0$  line are always parallel to the  $c$  axis at that point.

<sup>13</sup>A knife-edge case would represent a solution where, as if by “divine intervention”, the system will converge to a stable equilibrium only if the initial point happened to be on the saddle-path (and would instead diverge for any other initial condition).

<sup>14</sup>While in this specific case this is obviously not necessary, at least not in order to find the steady state (since the model can be solved in closed form as we just did in section 1.2.1), it will turn out to be quite helpful as it will allow us to introduce some of the concepts that we will be using later on to solve more complicated models in an easier settings.

the steady state). Hence, we will solve the system by approximating the evolution of consumption  $c(t)$  and capital  $k(t)$  at  $N$  discrete points in time  $t^n \in \{t^1, t^2, \dots, t^N\}$ . Let's also assume that the time grid is uniformly spaced, so that we can simply denote the distance between grid points by  $\Delta t$ , such that  $t^{n+1} = t^n + \Delta t$ .

By definition the time derivative of, say, capital is given by:

$$\dot{k}(t) \equiv \frac{\partial k(t)}{\partial t} = \lim_{h \rightarrow 0} \frac{k(t+h) - k(t)}{h}$$

and identically for consumption. This means that, if we choose a sufficiently small  $\Delta t$ , we can approximate this derivative as:

$$\dot{k}(t^n) \approx \frac{k(t^n + \Delta t) - k(t^n)}{\Delta t}. \quad (1.14)$$

Equation (1.14) is an example of a so-called **finite-difference** approximation; we will use such approximations extensively throughout this course.

Using the same approximation for  $\dot{c}$ , we can then write the (approximation of the) entire system of ODEs as:

$$\begin{aligned} \frac{k^{n+1} - k^n}{\Delta t} &= F(k^n) - \delta k^n - c^n \\ \frac{c^{n+1} - c^n}{\Delta t} \frac{1}{c^n} &= \frac{1}{\gamma(c^n)} (F'(k^n) - \delta - \rho) \end{aligned}$$

where we have used notation  $k^n = k(t^n)$  and  $c^n = c(t^n)$ . We can therefore rewrite this system as:

$$k^{n+1} = (F(k^n) - \delta k^n - c^n) \Delta t + k^n \quad (1.15)$$

$$c^{n+1} = \frac{c^n}{\gamma(c^n)} (F'(k^n) - \delta - \rho) \Delta t + c^n \quad (1.16)$$

and it is easy to see that, given a pair  $(k^n, c^n)$  we can immediately solve this system for  $(k^{n+1}, c^{n+1})$ . Hence, given any initial guess for  $(k^1, c^1)$ , we can solve this system of equations forward to obtain an approximated time path of consumption and capital.

Now, since the initial condition for capital is already given we can simply set  $k^1 = k_0$ , and all that is left to do is to ensure that we can find the initial level of consumption such that the initial guess  $(k^1, c^1)$  lies on the saddle path. If our initial level of consumption is correct and our approximation is sufficiently precise, the system should then converge to its unique steady-state.

In order to do that, we can use a so-called **shooting algorithm**, which essentially follows the exact same intuition we applied when analyzing the phase diagram in fig. 1.1 and is presented in algorithm 1.

We start by defining an arbitrarily small tolerance  $\varepsilon$ , we set  $k^1 = k_0$  (recall that  $k(0) = k_0$ ) and we guess  $c^1$ . Then we iterate forward the discretized system of eqs. (1.15) and (1.16) to find the path of consumption and capital,  $\{k^n, c^n\}_{n=1}^N$ , up to a sufficiently

---

**Algorithm 1** Shooting Algorithm

---

```

1: Define an arbitrarily small tolerance  $\varepsilon$  and set  $k^1 = k_0$ 
2: Guess  $c^1$ 
3: for  $n = 1$  to  $N$  do
4:   Find  $(k^{n+1}, c^{n+1})$  by solving

$$k^{n+1} = (F(k^n) - \delta k^n - c^n) \Delta t + k^n$$


$$c^{n+1} = \frac{c^n}{\gamma(c^n)} (F'(k^n) - \delta - \rho) \Delta t + c^n$$

5:   if  $k^n \leq 0$  or  $c^n \leq 0$  then
6:     Update  $c^1$  and go back to 2
7:   end if
8:   Compute the distance  $\varepsilon^n = \|(k^{n+1}, c^{n+1}) - (k^n, c^n)\|$ 
9:   if  $\varepsilon^n < \varepsilon$  then
10:    Exit
11:   end if
12: end for

```

---

large value of  $N$ . At each time-step  $n$  we check if the pair  $(k^n, c^n)$  violates either the transversality condition,  $c^n \leq 0$  – in which case we know the initial guess for consumption was too low and we need to increase it – or the feasibility condition,  $k^n \leq 0$  – in which case we know the initial guess for consumption was too high and we need to decrease it.

Finally, to evaluate convergence (i.e., to understand if we have reached the steady state) we can calculate the distance between two subsequent pairs as

$$\varepsilon^n = \|(k^{n+1}, c^{n+1}) - (k^n, c^n)\|$$

and check if it is smaller than our chosen tolerance  $\varepsilon^n < \varepsilon$ .<sup>15</sup>

---

<sup>15</sup>Note that we could in principle check this condition just for  $n = N$ . However, we generally gain in efficiency by checking at every time-step, because this way we can stop the iterations and update our initial guess as soon as one of the two conditions is violated. In fact, we can even do better than that and stop as soon as one of the two elements of the sequence starts decreasing (but not both); this is because, by studying the phase diagram, we know that it happens only off the equilibrium path.

# Chapter 2

## Deterministic HJB Equations

In this lecture we will continue using the neoclassical growth model as an example. However, we will use a different solution method that directly solves the Hamilton-Jacobi-Bellman (HJB) equation of the model (i.e. the recursive formulation of the sequential problem solved in the first lecture). We will therefore see how to define the HJB equation, as well as how to solve it numerically using a finite-differences (FD) approximation of the value function.

Just as in the first lecture, since this specific model can be solved in closed form (or numerically using a shooting algorithm as we did in chapter 1), writing down the HJB is not strictly necessary to solve the neoclassical growth model *per-se*. However, doing so will allow us to introduce concepts that we will be using later on to solve more complicated models in a familiar setting.

### 2.1. The Hamilton-Jacobi-Bellman Equation

As a first step, we will provide a “heuristic” approach on how to derive the HJB starting from any deterministic optimal control problem. As already introduced in chapter 1, such problem can be written as

$$v(x_0) = \max_{\alpha(t)} \int_0^{\infty} e^{-\rho t} r(x(t), \alpha(t)) dt \quad (2.1)$$

s.t.  $\dot{x}(t) = f(x(t), \alpha(t)).$

As we saw, the current-value Hamiltonian  $\mathcal{H}$  of this problem is:<sup>1</sup>

$$\mathcal{H}(x, \alpha, \lambda) = r(x, \alpha) + \lambda f(x, \alpha).$$

Shortly we will provide an alternative derivation of the **Hamilton-Jacobi-Bellman equation**; however, we can quickly get to it using an quick-and-dirty approach by using the above

---

<sup>1</sup>We drop the dependence on  $t$  to ease notation.

Hamiltonian as:

$$\rho v(x) = \max_{\alpha} r(x, \alpha) + v'(x)f(x, \alpha). \quad (2.2)$$

which makes the connection between the Hamiltonian and the HJB equations very apparent. First, the co-state  $\lambda$  in the Hamiltonian has been replaced by the derivative with respect to the state variable  $x$  of the value function,  $v'(x)$ , in the HJB:<sup>2</sup>

$$\lambda(t) = v'(x(t)).$$

Second, notice that if we write down the maximized Hamiltonian  $\hat{\mathcal{H}}(x, p)$

$$\hat{\mathcal{H}}(x, p) \equiv \max_{\alpha} r(x, \alpha) + pf(x, \alpha),$$

then the HJB equation can be written as:

$$\rho v(x) = \hat{\mathcal{H}}(x, v'(x)).$$

That is, the Hamilton-Jacobi-Bellman equation is given by the maximized Hamiltonian in which the co-state variable equals the derivative of the value function.<sup>3</sup>

Let's now apply the same approach to write the HJB of the neoclassical growth model. Recall that the Hamiltonian for that model is given by:

$$\mathcal{H}(k, c, \lambda) = u(c) + \lambda [F(k) - \delta k - c]$$

which means we can directly write the HJB equation as:

$$\rho v(k) = \max_c u(c) + v'(k) (F(k) - \delta k - c). \quad (2.3)$$

In order to solve the model, in section 2.2 we will see how to discretize the HJB eq. (2.3) directly to find the value function. However, with eq. (2.3) on hand, we can in principle follow the same steps as in discrete time to get to an Euler equation: take first order conditions, apply the envelope condition, and put them together to derive the Euler equation. To do so, we start by taking the derivative with respect to consumption to get the first order condition (FOC):

$$u'(c) = v'(k) \quad (2.4)$$

which, incidentally, also implies that  $\lambda = u'(c)$  just as in chapter 1. Then, we write the usual envelope condition:<sup>4</sup>

$$\rho v'(k) = v''(k) \underbrace{(F(k) - \delta k - c)}_k + v'(k) (F'(k) - \delta). \quad (2.5)$$

---

<sup>2</sup>As we will see in a second, this is in fact also going to be equal to the marginal utility of consumption, just as in chapter 1.

<sup>3</sup>Note also that, in this specific problem, the state variable is one dimensional. If this was not the case, in place of the derivative of the value function, we would have the gradient with respect to the state vector (and the dot product between the gradient and the law of motions for the different states).

<sup>4</sup>If you are not comfortable with the envelope theorem, see section B.1 for a quick review.

## 2.1. The Hamilton-Jacobi-Bellman Equation

---

In order to put the two eqs. (2.4) and (2.5) together, we can first differentiate both sides of eq. (2.4) with respect to time to obtain:

$$u''(c)\dot{c} = v''(k)\dot{k}.$$

Then, plugging this last equation in the envelope condition (2.5) we get:

$$\rho u'(c) = u''(c)\dot{c} + u'(c)(F'(k) - \delta)$$

which we can rearrange to obtain the exact same Euler equation that we got using the Hamiltonian approach:<sup>5</sup>

$$\frac{\dot{c}}{c} = -\frac{u'(c)}{u''(c)c}(F'(k) - \delta - \rho). \quad (2.6)$$

**Theoretical results** — To conclude this section it is useful to at least mention that, just as there are theoretical results about existence and uniqueness of the solution to discrete-time Bellman equations, there exist also similar results about the HJB equations in continuous time.<sup>6</sup> Since a treatment of these theoretical results is far beyond the scope of these notes, we hereby just briefly mention them:

**Theorem 1:** *The HJB eq. (2.2) has a unique “nice” solution.*

**Theorem 2:** *The solution to the HJB eq. (2.2) equals the value function in eq. (2.1).*

While in this version these theorems are extremely vague (and deliberately so), they are still very important because they ensure that the we can find the solution we are looking for: on the one hand, the first theorem makes sure that the solution to the HJB is unique and has certain desirable properties – specifically, by “nice” we mean that the solution to the HJB is a viscosity solution;<sup>7</sup> on the other, the second theorem ensures us that, by finding the solution to the recursive problem, we also find the solution to our problem of interest (namely, the sequential problem).

### 2.1.1. An Alternative Derivation

An alternative way to derive the HJB equation which does not make use of the Hamiltonian instead involves simply start from its discrete-time counterpart. While this approach has the advantage of being more intuitive for people used to working with discrete-time models, it may also lead to incorrect results if one is not very careful (especially once we will introduce uncertainty in chapter 3) In any case, since it can also be quite instructive, it is useful to do it at least once.

---

<sup>5</sup>The Euler Equation (2.6) is obviously identical to eq. (1.8) derived in chapter 1, so we could in theory use the same solution method here.

<sup>6</sup>For a review of such results in discrete time, see Stokey et al. 1989, Chapter 4.

<sup>7</sup>For an introduction to the concept of viscosity solution, the interested reader can refer to Appendix D of Achdou et al. 2022, and the references therein.

Let's start by working with the discrete-time model and by assuming a time period of length  $\Delta$ .<sup>8</sup> We also assume that the discount factor is in the exponential form  $\beta(\Delta) = e^{-\rho\Delta}$ . The discrete-time Bellman equation is given by:<sup>9</sup>

$$\begin{aligned} v(k_t) &= \max_{c_t} \Delta u(c_t) + e^{-\rho\Delta} v(k_{t+\Delta}) \\ \text{s.t. } k_{t+\Delta} &= \Delta(F(k_t) - \delta k_t - c_t) + k_t \end{aligned}$$

which obviously boils down to the standard Bellman equation if we set  $\Delta = 1$ .

For small values of  $\Delta$ , we can approximate the discount factor as:

$$e^{-\rho\Delta} \approx 1 - \rho\Delta$$

and, as a consequence, the HJB can also be approximated as:

$$v(k_t) \approx \max_{c_t} \Delta u(c_t) + (1 - \rho\Delta) v(k_{t+\Delta}).$$

Subtracting  $(1 - \rho\Delta)v(k_t)$  and dividing by  $\Delta$  on both sides leads to

$$\rho v(k_t) \approx \max_{c_t} u(c_t) + u(c_t) + (1 - \rho\Delta) \frac{v(k_{t+\Delta}) - v(k_t)}{\Delta}.$$

Finally, multiply and divide the last term on the right-hand side by  $k_{t+\Delta} - k_t$  to obtain:

$$\rho v(k_t) \approx \max_{c_t} u(c_t) + (1 - \rho\Delta) \frac{v(k_{t+\Delta}) - v(k_t)}{k_{t+\Delta} - k_t} \frac{k_{t+\Delta} - k_t}{\Delta}$$

which, taking the limit for  $\Delta \rightarrow 0$ , gives us the HJB equation as in eq. (2.3):

$$\rho v(k_t) = \max_{c_t} u(c_t) + v'(k_t) \dot{k}_t$$

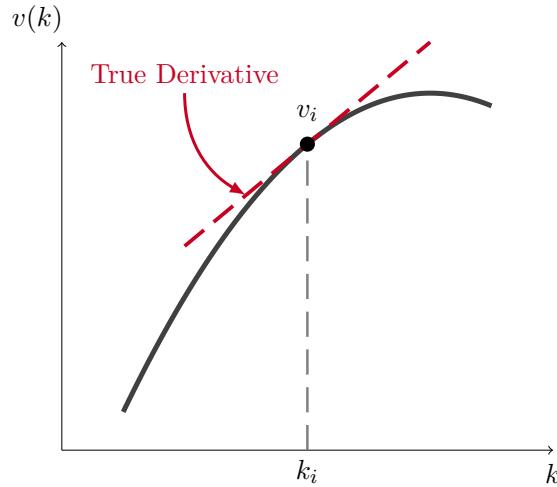
## 2.2. Numerical Solution

Except in some specific cases it is often impossible to find an analytical solution to the HJB equation. However, in the same way that there exists a well-developed theory for the *theoretical* properties of HJB equations, there also exists a well-developed theory on how to find the *numerical* solution of the HJB. In particular, this theory tells us that if we define an “appropriate” approximation scheme for the HJB, such scheme will converge to the unique viscosity solution.<sup>10</sup> In terms of the approximation scheme, we will focus on so-called **finite-difference** (FD) methods, which are arguably the simplest and most

<sup>8</sup>Eventually, we will let  $\Delta$  go to zero to get the continuous-time limit of our model.

<sup>9</sup>We're changing notation with respect to time by writing  $x(t)$  as  $x_t$  to stay consistent with the usual discrete time notation.

<sup>10</sup>In order for it to converge to the viscosity solution, the “appropriately” defined approximation scheme needs to satisfy the following three main conditions: monotonicity, consistency, and stability. Of these three conditions, consistency and stability are usually easily satisfied; while monotonicity is sometimes a bit more challenging to prove. See Barles and Souganidis (1991) for more details.



**Figure 2.1:** True Derivative

easily applicable.<sup>11</sup> However, it is worth mentioning that there exist also other well-developed methods (used mostly in other disciplines) such as finite-element, finite-volume, and spectral methods, among many others.

### 2.2.1. Finite-differences

When we were looking for a solution to the neoclassical growth model in chapter 1 we wrote down the system of ODEs that characterized our system and discretized it along the time dimension. However, in our HJB eq. (2.3) – which we rewrite here for convenience:

$$\rho v(k) = \max_c u(c) + v'(k) (F(k) - \delta k - c)$$

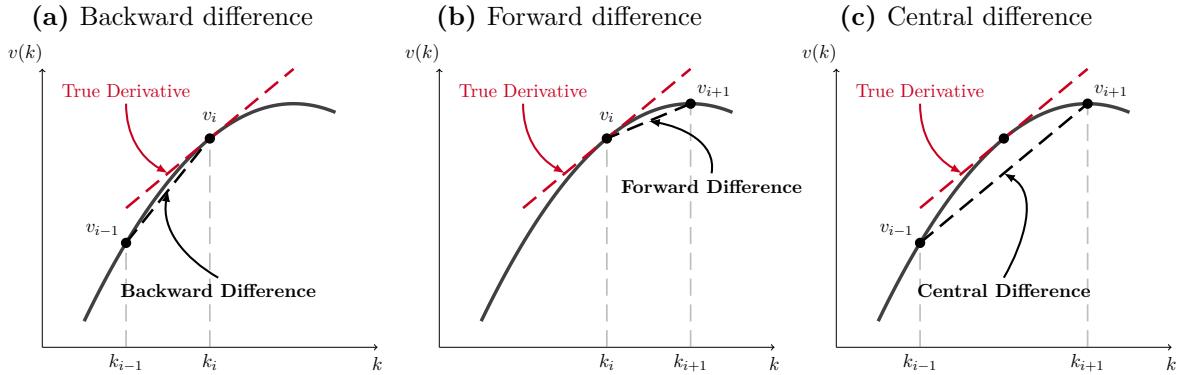
we can immediately observe that the value function depends on capital, and not on time. As a consequence, when looking for a solution to eq. (2.3), it seems natural to want to discretize the state space of our value function, i.e. capital  $k$ . In particular, since the HJB equation also includes the derivative of the value function with respect to capital, we will need to approximate that derivative. As we will see, much of the discussion in the rest of this chapter will be about how to define such an approximation in an “appropriate” way.

We start by defining a grid for capital  $\{k_i\}_{i=1}^I$ . Approximating the value function  $v(k)$  itself is rather trivial because it suffices to evaluate it at each discrete point on the grid  $k_i$ . Approximating the derivative  $v'(k)$  requires a bit more thought. In fact, if we want to find such an approximation by using our approximated value function, at least three options come to mind:<sup>12</sup>

- **Backward difference:** We approximate the derivative at point  $k_i$  by using the ap-

<sup>11</sup>A good entry-level reference for finite-difference methods in economics and finance is Tourin (2013).

<sup>12</sup>We hereby use the shorthand notation  $v_i \equiv v(k_i)$



**Figure 2.2:** Finite Difference Approximations

proximated value function at points  $k_i$  and  $k_{i-1}$  (see fig. 2.2a):

$$v'_{i,B} = \frac{v_i - v_{i-1}}{\Delta k} \approx v'(k_i)$$

- **Forward difference:** We approximate the derivative at point  $k_i$  by using the approximated value function evaluated at points  $k_i$  and  $k_{i+1}$  (see fig. 2.2b):

$$v'_{i,F} = \frac{v_{i+1} - v_i}{\Delta k} \approx v'(k_i)$$

- **Central difference:** We approximate the derivative at point  $k_i$  by using the value function evaluated at points  $k_{i-1}$  and  $k_{i+1}$  (see fig. 2.2c):

$$v'_{i,C} = \frac{v_{i+1} - v_{i-1}}{2\Delta k} \approx v'(k_i)$$

These three approaches are justified by the fact that we can easily evaluate the value function at point  $k_i$  – assuming we know it – and then, in order to approximate the derivative at that same point, we simply “move away” from  $v_i$  in a given direction. Because we can in principle move backward, forward, or both we have the above three choices. Figure 2.2 shows graphically how these three approaches work.

Of course our derivative approximation will get more and more accurate as the grid gets finer and finer (i.e., with smaller  $\Delta k$ ); however, this obviously comes at the cost of having to evaluate the value function at more points. While we certainly want to find a scheme that converges to the true solution as the grid gets finer, we are also particularly interested in finding a scheme that allows us to have a better approximation of the true derivative *conditional* on the size of the grid.<sup>13</sup>

---

<sup>13</sup>Since we mentioned grid density, it might be worth noting that, if we know that the value function is going to have more curvature in some specific regions of the state space, it is often useful to use a non-uniform grid that concentrates more points in that region. In order to make the exposition simpler, throughout these notes we will always be using uniform grids; however, most results apply rather straight-

Applying our finite-difference approximations to the HJB eq. (2.3) we have:

$$\rho v_i = u(c_i) + v'_i \cdot (F(k_i) - \delta k_i - c_i) \quad (2.7)$$

where  $c_i$  can be “extracted” from the FOC of our problem:  $c_i = (u')^{-1}(v'_i)$  and  $v'_i$  is the approximated derivative of the value function  $v'(k)$  at  $k_i$  obtained using one of backward, forward or central FD approximations.

There are essentially two problems that we will need to solve in order to find a solution to eq. (2.7): The first one is related to which approximation scheme we should use. The finite difference scheme that we choose needs to satisfy certain monotonicity, consistency, and stability conditions to guarantee convergence of the numerical solution to (2.7) to the true viscosity solution to (2.3). Our chosen approximation scheme will be the so-called **upwind scheme**, which we cover in section 2.2.2.

The second problem that we face is related to the non-linearity of the HJB equation. Such non-linearity is made apparent for example by the fact that  $c_i = (u')^{-1}(v'_i)$ . Essentially, in eq. (2.7)  $v_i$  is defined as a non-linear function of  $v'_i$ , which is itself however defined as a function of  $v_i$ . We will solve this problem by using an iterative scheme, similarly to what we did in the shooting algorithm of chapter 1: we will start with a guess for the value function, which will allow us to compute its derivative and, as a consequence, also consumption. These two together will allow us to update our guess for the value function and to iterate this procedure until it hopefully converges. As we will see there are essentially two ways of doing this: by using an explicit or an implicit method.<sup>14</sup> While each has its own advantages and disadvantages, generally speaking, for these type of heterogeneous-agent models the advantages of using an **implicit method** far outweigh its possible disadvantages, so we will mostly be focusing on that in section 2.2.4.

### 2.2.2. Upwind Scheme

The upwind scheme essentially boils down to cleverly choosing when to use the backward- or the forward-difference to approximate the derivative of the value function. As we will see shortly, this choice will be based on the sign of the **drift** of the state variable. In particular, the upwind scheme will use the forward-difference when the drift is positive and the backward-difference when it is negative. However, while this sounds straightforward in theory, there is one major caveat: the drift itself also depends on which approximation we choose. To see this, recall that the drift of capital is given by:

$$\dot{k} = F(k) - \delta k - c,$$

and that from the FOC we have that  $c = (u')^{-1}(v')$ . Hence consumption (and therefore  $\dot{k}$ ) is a function of the derivative  $v'$ . Because of that, we already have two ways of defining

---

forwardly in the case of non-uniform grids. In section B.2 we show one way of creating such a non-uniform grid.

<sup>14</sup>While we ignored this issue in our earlier treatment of the shooting algorithm, the same distinction could have been made there as well.

the drift of our state variable: using the forward difference or the backward difference; since the sign of the drift might differ depending on which approximation we use, one has to be a bit careful in the way the scheme is designed.

Let's start by defining the drift of the state variable in these two cases as:

$$s_{i,F} = F(k_i) - \delta k_i - (u')^{-1}(v'_{i,F})$$

$$s_{i,B} = F(k_i) - \delta k_i - (u')^{-1}(v'_{i,B})$$

which for brevity we will refer to as the forward-drift and the backward-drift, respectively (we use the letter  $s$  for savings).

As we just said, the **upwind scheme** tells us to:

- ▶ Use the forward difference if the drift is positive
- ▶ Use the backward difference if the drift is negative

Intuitively, if the drift is positive, it means that our state variable is increasing and the derivative of our function of interest is likely better approximated using the forward difference (and viceversa if the drift is negative); that is, we use the derivative approximation in the direction in which the state is moving.

In the above definition of the upwind scheme, however, we have not specified whether we refer to the backward- or the forward-drift. In most cases, as long as our grid is sufficiently dense, these two should be sufficiently close to each other as to have the same sign. However, there might be cases in which they are not. To account for all possible cases, what we do is calculate both the backward- and forward-drift,  $s_{i,B}$  and  $s_{i,F}$ , and assess their sign:

- ▶ If both are positive

$$s_{i,F} > 0 \quad \text{and} \quad s_{i,B} \geq 0,$$

then we approximate  $v'$  using the forward derivative:

$$\dot{k} > 0 \implies v'_i = v'_{i,F}, \quad c_{i,F} = (u')^{-1}(v'_{i,F}), \quad s_i = s_{i,F}.$$

- ▶ If both are negative

$$s_{i,F} \leq 0 \quad \text{and} \quad s_{i,B} < 0$$

then we approximate  $v'$  using the backward derivative:

$$\dot{k} > 0 \implies v'_i = v'_{i,B}, \quad c_{i,B} = (u')^{-1}(v'_{i,B}), \quad s_i = s_{i,B}$$

- ▶ Finally, if the forward drift is negative and the backward drift is positive

$$s_{i,F} \leq 0 \quad \text{and} \quad s_{i,B} \geq 0$$

then, intuitively, we know we should be at a “steady-state” (really, a maximum): if we were to use the forward drift it would tell us to “go backward” and if we were to

## 2.2. Numerical Solution

---

use the backward drift it would tell us to “go forward”. Since we are at steady-state, our state variable is not moving, i.e.  $\dot{k} = 0$ , and our drift is zero

$$\dot{k} = 0 \implies s_i = 0, \quad c_{i,0} = F(k_i) - \delta k_i, \quad v'_i = u'(F(k_i) - \delta k_i)$$

that is, capital is constant, which means we can use the law of motion for capital to obtain the value of consumption and then find our approximation of the derivative of the value function directly from the FOC.

To recap, our approximation of the derivative of the value function will be given by the forward difference if the forward-drift is positive, by the backward difference if the backward-drift is negative, and by its steady-state value when the two drifts have opposite signs. In shorter terms:

$$v'_i = v'_{i,F} \cdot \mathbf{1}_{\{s_{i,F} > 0\}} + v'_{i,B} \cdot \mathbf{1}_{\{s_{i,B} < 0\}} + \bar{v}'_i \cdot \mathbf{1}_{\{s_{i,F} \leq 0 \leq s_{i,B}\}}$$

where  $\bar{v}'_i = u'(F(k_i) - \delta k_i)$  and  $\mathbf{1}_{\{\cdot\}}$  is the indicator function.<sup>15</sup>

The careful reader might have noticed that we did not cover the case when the forward-drift is positive *and* the backward-drift is negative. In general, as long as the value function  $v$  is increasing and concave, this should never happen. In fact, if  $v$  is concave, we will generally have  $v'_{i,F} < v'_{i,B}$  (e.g. compare figs. 2.2a and 2.2b) which implies that the forward drift will generally be less than the backward drift:  $s_{i,F} < s_{i,B}$ .<sup>16</sup> In principle, however, if the value function is (locally or globally) convex at  $k_i$  it could be the case that  $s_{i,F} > 0$  and  $s_{i,B} < 0$ .<sup>17</sup> This obviously begs the question of what to do in this case. One of the nicest properties about the upwind scheme is that it works also in this case (with the appropriate adjustments); all we have to do is to choose in which “direction” to go by looking at the value of the Hamiltonian instead of looking at the sign of the drift. Specifically, after

---

<sup>15</sup>When searching for references on the upwind scheme, one might find it to be defined the other way around; that is, by using the derivative on the side *opposite* to that of the drift (or flow) of a variable. The reason for this discrepancy is in the direction of time. As we will see in the coming lectures, the HJB is a *backward* equation in time; that is, it can be solved starting from a terminal condition and moving back in time. Because the bias in the upwind scheme needs to change side based on the direction of time, it is defined differently for backward and forward equations (and most general treatments usually start from forward equations).

<sup>16</sup>This is due to the fact that marginal utility is decreasing:

$$(u')^{-1}(v'_{i,F}) > (u')^{-1}(v'_{i,B})$$

which, when calculating the drifts, implies:

$$s_{i,F} = F(k_i) - \delta k_i - (u')^{-1}(v'_{i,F}) < s_{i,B} = F(k_i) - \delta k_i - (u')^{-1}(v'_{i,B})$$

<sup>17</sup>Since we will always be approximating  $v(k)$  with  $v_i$ , it is actually possible that along the convergence path our approximation becomes locally convex even if the true objective function is concave.

defining the forward Hamiltonian,  $H_{i,F}$ , and the backward Hamiltonian,  $H_{i,B}$ , as

$$H_{i,F} \equiv u(c_i) + v'_{i,F} [F(k_i) - \delta k_i - c_i] \quad (2.8)$$

$$H_{i,B} \equiv u(c_i) + v'_{i,B} [F(k_i) - \delta k_i - c_i] \quad (2.9)$$

the upwind scheme can be adjusted by using the forward difference if the forward Hamiltonian is greater than the backward Hamiltonian and the backward difference in the opposite case (however, one also needs to check if setting the drift to zero gives a higher Hamiltonian). In section B.3 we look at one example featuring non-convexities in which this distinction is particularly important.

### 2.2.3. Discretized HJB Equation

Now that we have an approximation of the derivative of the value function, we can also write the approximation of the HJB equation:

$$\rho v_i = u(c_i) + \frac{v_{i+1} - v_i}{\Delta k} \cdot s_{i,F} \cdot \mathbf{1}_{\{s_{i,F} > 0\}} + \frac{v_i - v_{i-1}}{\Delta k} \cdot s_{i,B} \cdot \mathbf{1}_{\{s_{i,B} < 0\}}$$

where  $c_i = c_{i,F} \cdot \mathbf{1}_{\{s_{i,F} > 0\}} + c_{i,B} \cdot \mathbf{1}_{\{s_{i,B} < 0\}} + c_{i,0} \cdot \mathbf{1}_{\{s_{i,F} \leq 0 \leq s_{i,B}\}}$  is computed using the same approximation of the derivative, and we have not included the third case ( $s_{i,F} \leq 0$  and  $s_{i,B} \geq 0$ ) on the right-hand side because in that case  $s_i = 0$  by construction. The approximated HJB can also be written, in shorter notation, as:

$$\rho v_i = u(c_i) + v'_{i,F} \cdot s_{i,F}^+ + v'_{i,B} \cdot s_{i,B}^- \quad (2.10)$$

where  $x^+ = \max\{x, 0\}$  and  $x^- = \min\{x, 0\}$ .

It is at this point convenient to write the discretized HJB equation in matrix form as:

$$\rho \mathbf{v} = \mathbf{u} + \mathbf{A} \mathbf{v} \quad (2.11)$$

where  $\mathbf{v}$  and  $\mathbf{u}$  are  $I$ -dimensional column vectors,  $\mathbf{A}$  is an  $I \times I$  matrix, and  $I$  is the number of points in the discretized state space. Zooming in on the  $i^{th}$  row of  $\mathbf{A}$  and the related components of the  $\mathbf{v}$  vector, we have that the only non-zero elements of the  $i^{th}$  row are in columns  $i-1$ ,  $i$ , and  $i+1$ :

$$\underbrace{\begin{pmatrix} -\frac{s_{i,B}^-}{\Delta k} & \underbrace{\frac{s_{i,B}^-}{\Delta k} - \frac{s_{i,F}^+}{\Delta k}}_{\text{outflow}_i \leq 0} & \frac{s_{i,F}^+}{\Delta k} \\ \text{inflow}_{i-1} \geq 0 & & \text{inflow}_{i+1} \geq 0 \end{pmatrix}}_{i^{th} \text{ row of } \mathbf{A}} \begin{pmatrix} v_{i-1} \\ v_i \\ v_{i+1} \end{pmatrix}.$$

It is easy to see that the matrix representation in eq. (2.11) is equivalent to the scalar one

in eq. (2.10) by performing the matrix multiplication on the right-hand side:<sup>18</sup>

$$\begin{aligned}\mathbf{A}_{i,i-1}v_{i-1} + \mathbf{A}_{i,i}v_i + \mathbf{A}_{i,i+1}v_{i+1} &= -\frac{s_{i,B}^-}{\Delta k}v_{i-1} + \left( \frac{s_{i,B}^-}{\Delta k} - \frac{s_{i,F}^+}{\Delta k} \right)v_i + \frac{s_{i,F}^+}{\Delta k}v_{i+1} \\ &= s_{i,B}^- \frac{v_i - v_{i-1}}{\Delta k} + s_{i,F}^+ \frac{v_{i+1} - v_i}{\Delta k} \\ &= v'_{i,F} \cdot s_{i,F}^+ + v'_{i,B} \cdot s_{i,B}^-. \end{aligned}$$

The  $\mathbf{A}$  matrix has several properties that it is useful to make note of:

- ▶ Its rows sum to zero.
- ▶ It is tridiagonal; that is, its only nonzero elements are on the main diagonal, the first lower diagonal (the first diagonal below the main diagonal), and the first upper diagonal (the first diagonal above the main diagonal). Importantly, this also implies that  $\mathbf{A}$  is going to be an extremely **sparse matrix**.<sup>19</sup>
- ▶ It has negative elements on the main diagonal and positive elements off-diagonal. In particular recall that  $s_{i,F}^+ = \max\{s_{i,F}, 0\} \geq 0$  and  $s_{i,B}^- = \max\{s_{i,B}, 0\} \leq 0$ .
- ▶ It contains information about flows from point  $i$  to point  $i-1$  and  $i+1$ , not too differently from a transition matrix in discrete time. These essentially tell us the rate at which households are moving out of point  $i$  towards either  $i-1$  and/or  $i+1$ .

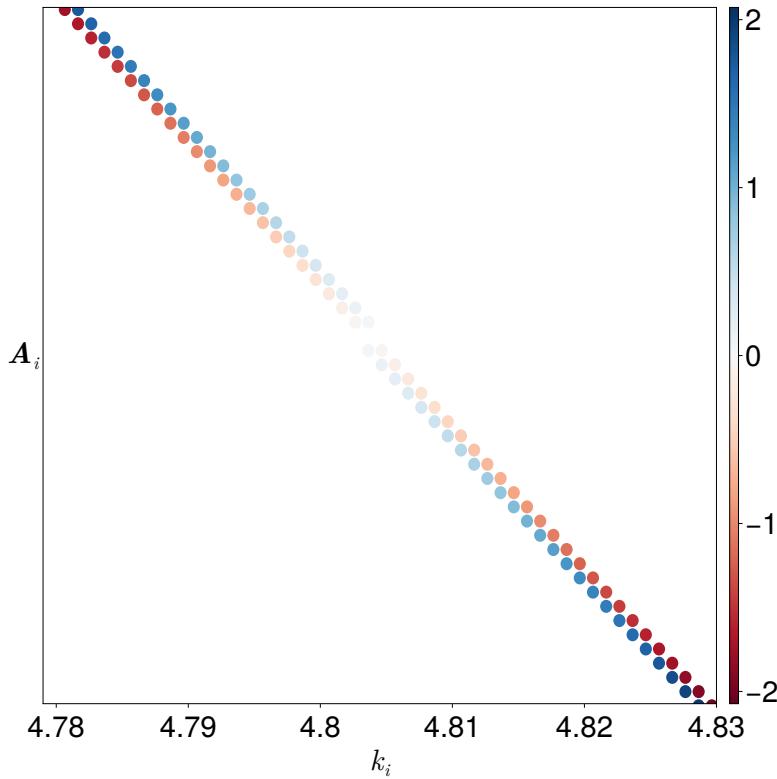
Figure 2.3 presents a visual representation of the  $\mathbf{A}$  matrix for the neoclassical growth model, where each dot represents a non-zero element of  $\mathbf{A}$ , with lighter dots corresponding to smaller entries (in absolute value). By analyzing this figure we can see that  $\mathbf{A}$  is extremely sparse, that it is tri-diagonal and that the upwind scheme is selecting the forward difference at the low end of the state space (in the top-left of  $\mathbf{A}$  there are only elements on the upper and main diagonals) and the backward derivative at the top end of the state space (in the bottom-right of  $\mathbf{A}$  there are only elements on the lower and main diagonals). Finally, the colors of the dots also show us that inflows and outflows get larger farther away from the steady state and closer to zero at the steady state (the empty spot at the center of the matrix).

Before moving on to how to handle the non-linearity of eq. (2.7), there is one final issue that needs to be at least mentioned: by construction the upwind scheme uses either the backward difference or the forward difference based on the sign of the drift; however, these are not well defined at the boundaries of the state space. In fact, we cannot really define a backward difference at the first point of our grid without a point to the left of it, and similarly we cannot define a forward difference at the last point of our grid without a point to the right of it (these are sometimes called *ghost nodes*). For the time being, this is actually not going to be an issue since the upwind scheme automatically chooses

---

<sup>18</sup>We are here making use of the fact that  $\mathbf{A}$  is tridiagonal and  $\mathbf{A}_{i,j} = 0 \forall j < i-1, j > i+1$ .

<sup>19</sup>In this specific model, out of the  $I^2$  total elements in  $\mathbf{A}$ , *at most*  $3I-2$  will be non-zero. Hence, if we were to discretized capital on a grid with  $I = 1,000$  points, that would mean that less than 0.3% of the entries of  $\mathbf{A}$  would be non-zero.



**Figure 2.3:** Visualization of the  $\mathbf{A}$  matrix for the neoclassical growth model

the forward difference for  $i = 1$  and the backward difference for  $i = I$  by virtue of the fact that, due to the concavity of our problem, the drift is positive at the beginning of the state space and negative at its end (as long as we choose a sufficiently large  $k_I$ ). This also makes economic sense: for low level of capital, the household will want to save,  $\dot{k} > 0$ , and for high level of capital, she will want to dissave,  $\dot{k} < 0$ . However, this is simply a property of the model we are solving and not of the upwind scheme itself. In fact, when we will go to different models in later lectures, we will have to impose so-called *boundary conditions* which deal specifically with what happens at the boundaries of the state space. We defer the discussion of boundary conditions to section 4.2.

## 2.2.4. Explicit and Implicit Methods

Until now, we have deliberately ignored the non-linearity of the HJB equation. In fact, to be precise, both  $\mathbf{A}$  and  $\mathbf{u}$  are functions of  $\mathbf{v}$  itself: because  $s_{i,B}^-$  and  $s_{i,F}^+$  are functions of  $\mathbf{v}$ ,  $\mathbf{A}$  will also be a function of it, while  $\mathbf{u}$  is clearly related to  $\mathbf{v}$  through the FOC,  $c = (u')^{-1}(v')$ . As a consequence, a more precise definition of our approximated HJB would be:

$$\rho\mathbf{v} = \mathbf{u}(\mathbf{v}) + \mathbf{A}(\mathbf{v})\mathbf{v}.$$

This means that we cannot generally solve for  $\mathbf{v}$  in “one shot”, but we will need to iterate on this equation in order to find a fixed point.

**Algorithm 2** Explicit Method

---

Define an arbitrarily small tolerance  $\varepsilon$  and initial guess  $\mathbf{v}^0$

**for**  $n = 0, 1, 2, \dots$  **do**

    Given a guess  $v_i^n$ , find  $v_i^{n+1}$  by solving

$$\frac{v_i^{n+1} - v_i^n}{\Delta} + \rho v_i^n = u(c_i^n) + (v_i^n)' \cdot (F(k_i) - \delta k_i - c_i^n) \quad (2.12)$$

Where  $(v_i^n)'$  is the FD approximation to  $v'(k_i)$  and  $c_i^n = (u')^{-1}[(v_i^n)']$

**if**  $\|\mathbf{v}^{n+1} - \mathbf{v}^n\| < \varepsilon$  **then**

        Exit

**else**

        Set  $n = n + 1$  and go back

**end if**

**end for**

---

In order to do so, we can use one of two options:

- ▶ An **explicit method**, which is easier to write down and in which each iteration is faster to solve, but also less stable and inefficient
- ▶ An **implicit method**, which is slightly harder to write down and in which each iteration is slower, but is also more stable and efficient.

Both methods boil down to iterating on  $n$  in the following equation:

$$\frac{v_i^{n+1} - v_i^n}{\Delta} + \rho v_i^\bullet = u(c_i^n) + (v_i^\bullet)' \cdot (F(k_i) - \delta k_i - c_i^n)$$

which simply adds a convergence term on the left-hand side that hopefully goes to zero as  $n$  grows and ensures that we have reached a fixed point of eq. (2.7).<sup>20</sup> The difference between the explicit and implicit methods will simply be in what  $v_i^\bullet$  is, where the explicit method sets  $v_i^\bullet = v_i^n$  and the implicit method  $v_i^\bullet = v_i^{n+1}$ . As we will see, for these types of models the implicit method is generally a much better choice; however, since the explicit method is slightly more intuitive we cover it first.

**Explicit method —** The explicit method is formalized in algorithm 2. A few things worth noting:

1. The step-size along the “time” dimension,  $\Delta$ , need not be (and in fact it should not be) the same as the step-size of the state-space discretization  $\Delta_k$
2. Equation (2.12) is **not** a contraction mapping, which means that nothing ensures us the algorithm will converge to the true solution

<sup>20</sup>We can in fact think of these methods as being equivalent to time-iteration; that is, solving a discretized version of:

$$\dot{v} + \rho v = u + \mathcal{A}v$$

where  $\mathcal{A}$  is the infinitesimal generator of the HJB (which we will define later).

---

**Algorithm 3** Implicit Method

Define an arbitrarily small tolerance  $\varepsilon$  and initial guess  $\mathbf{v}^0$

**for**  $n = 0, 1, 2, \dots$  **do**

    Given a guess  $v_i^n$ , find  $v_i^{n+1}$  by solving

$$\frac{v_i^{n+1} - v_i^n}{\Delta} + \rho v_i^{n+1} = u(c_i^n) + (v_i^{n+1})' \cdot (F(k_i) - \delta k_i - c_i^n) \quad (2.13)$$

Where  $(v_i^n)'$  is the FD approximation to  $v'(k_i)$  and  $c_i^n = (u')^{-1}[(v_i^n)']$

Solve the linear system:

$$\frac{1}{\Delta} (\mathbf{v}^{n+1} - \mathbf{v}^n) + \rho \mathbf{v}^{n+1} = \mathbf{u}^n + \mathbf{A}_n \mathbf{v}^{n+1}$$

↓

$$\mathbf{v}^{n+1} = \left( \left( \rho + \frac{1}{\Delta} \right) \mathbf{I} - \mathbf{A}_n \right)^{-1} \left( \mathbf{u}^n + \frac{1}{\Delta} \mathbf{v}^n \right)$$

**if**  $\|\mathbf{v}^{n+1} - \mathbf{v}^n\| < \varepsilon$  **then**

    Exit

**else**

    Set  $n = n + 1$  and go back to step 2

**end if**

**end for**

---

3. Related to the above two points: in order for it to converge, the explicit method usually requires a step-size  $\Delta$  that is sufficiently small.<sup>21</sup>

**Implicit method** — The implicit method is formalized in algorithm 3, which we can immediately notice is very similar to algorithm 2, the only difference being that we have carefully selected where to substitute our current guess  $v_i^n$  with the next guess  $v_i^{n+1}$ .<sup>22</sup> Compared to the explicit method, a few differences are worth noting:

1. While eq. (2.12) is of the form  $\mathbf{x} = \mathbf{Ab}$  and can therefore be solved at the expense of a single matrix multiplication, eq. (2.13) is of the form  $\mathbf{Ax} = \mathbf{b}$  and requires a matrix inversion to be solved (which is obviously more complicated).
2. Related to the above, numerical softwares can usually take advantage of sparsity to ensure the matrix inversion is done efficiently. This means that, since  $\mathbf{A}$  is ex-

---

<sup>21</sup>In the explicit method, one can in fact derive an expression for the maximum possible  $\Delta$  that can be used as a function of  $\Delta_k$ . Suffice here to say that, the finer is our state-space approximation (i.e. the smaller the  $\Delta_k$ ), the smaller our  $\Delta$  will need to be to ensure convergence.

<sup>22</sup>In fact algorithm 2 is technically only a *semi*-implicit method, since a fully-implicit scheme would also replace  $c_i^n$  with  $c_i^{n+1}$ . However, doing so would imply that we would need to solve a non-linear system of equations at each iteration. On the other hand the semi-implicit scheme defined in algorithm 3 retains many of the useful convergence properties of a fully-implicit method while preserving the linearity of the system. Because at each iteration we now have to solve a system of *linear* equations, this means the semi-implicit method is often faster (and potentially numerically more stable).

tremely sparse, even the system of the form  $\mathbf{Ax} = \mathbf{b}$  can be solved in a fraction of a second.<sup>23,24</sup>

3. The implicit method is stable “independently” of  $\Delta$ , which means we can choose a much larger value of  $\Delta$  than in the implicit method.

Effectively the sparsity of  $\mathbf{A}$  means that, while a single iteration of the implicit scheme takes marginally longer than a single iteration of the explicit scheme, the possibility to choose a larger step-size  $\Delta$  (i.e., larger “convergence steps”) means that the system needs a lot fewer steps to converge. As a consequence, while slightly more complicated the implicit method will generally be a lot faster (and more stable) than the explicit method.

**Explicit vs. Implicit approximations** — To have a better grasp of the differences between the two schemes, let’s briefly consider an easier example in the following first-order linear ODE

$$\dot{y}(t) = -\alpha y(t) \quad (2.14)$$

with initial condition  $y(0) = 1$ . Due to its simplicity we can easily find the analytical solution, which is simply given by:

$$y(t) = e^{-\alpha t}$$

However, even if we know its true solution, we can still approximate this equation numerically using either an explicit or an implicit method just as we did with the HJB.

In both cases we can approximate the time derivative on the left-hand side of eq. (2.14) as

$$\dot{y}(t) \approx \frac{y(t + \Delta) - y(t)}{\Delta},$$

while the difference is going to be in how we handle the RHS (i.e. whether we use  $y(t)$  or  $y(t + \Delta)$ ): with an explicit scheme, we use  $y(t)$  on the RHS; with an implicit scheme, we use  $y(t + \Delta)$ .<sup>25</sup>

With the explicit scheme we therefore have:

$$\frac{y(\Delta) - y(0)}{\Delta} = -\alpha y(0)$$

which rearranging and using the initial condition  $y(0) = 1$  becomes:

$$y(\Delta) = (1 - \alpha\Delta) \quad (2.15)$$

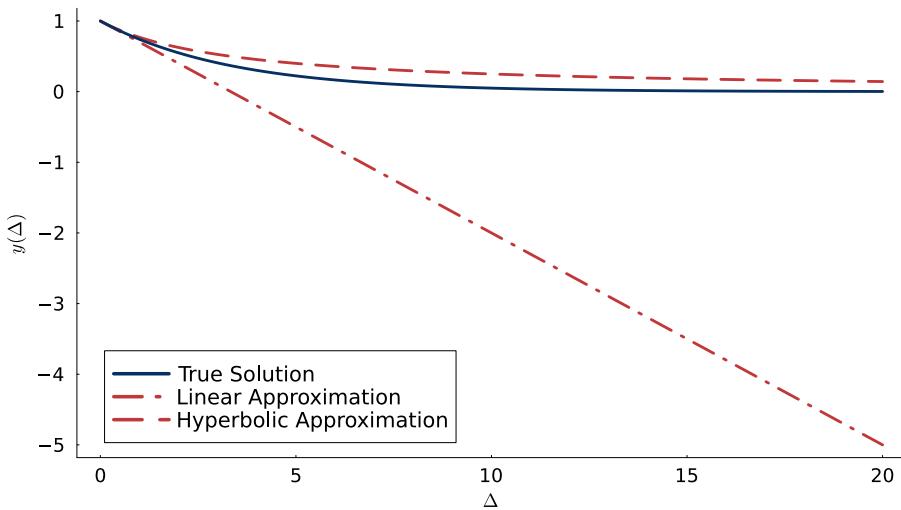
that is, we are approximating  $y(t)$  using a linear approximation.

---

<sup>23</sup>In order to take advantage of the sparsity of  $\mathbf{A}$  most numerical software generally require it to be *explicitly* declared as sparse (sparse matrices are often stored and handled differently than full matrices).

<sup>24</sup>Note that in most numerical softwares you don’t even need to perform the matrix inversion *per-se*, as they often include routines to solve system of equations more efficiently (this is usually done with a so-called *backslash* operator so that the system  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$  can be solved directly as  $\mathbf{x} = \mathbf{A}\backslash\mathbf{b}$ )

<sup>25</sup>Finally, since we already have  $y(0)$  given, it will make sense to set  $t = 0$  in both the implicit and explicit schemes.



**Figure 2.4:** Explicit and Implicit methods' comparison

On the other hand if we use the implicit method on the right hand side we have  $y(\Delta)$  and we have:

$$\frac{y(\Delta) - y(0)}{\Delta} = -\alpha y(\Delta)$$

which rearranging and using the initial condition leads to:

$$y(\Delta) = \frac{1}{1 + \alpha\Delta} \quad (2.16)$$

that is, we are approximating  $y(t)$  using a hyperbolic approximation.

Figure 2.4 shows how well the two approximations in eqs. (2.15) and (2.16) compare to the true solution in eq. (2.14) for different values of  $\Delta$ . It should be immediately apparent that, while the linear approximation (i.e., the explicit scheme) is only good for values of  $\Delta$  that are sufficiently small, the hyperbolic approximation (i.e., the implicit scheme) remains much closer to the true solution also for very large  $\Delta$ .

### 2.2.5. Results

Now that we know how to write down the HJB, approximate it with finite-difference method using the upwind scheme, and find the solution using an implicit method we can quickly look at the results. Since the neoclassical growth model is fairly well known, rather than focusing on the economics we will simply look at the numerical properties of the solution and make sure that it indeed coincides with the solution we obtained using the shooting algorithm in chapter 1.

In terms of model specifications we assume that the household has CRRA utility with risk aversion coefficient of 2 and discounts the future at 5% rate, the production function is  $F(k) = k^\alpha$  with  $\alpha = 0.3$ , and the depreciation rate  $\delta$  is set to 5%. We solve the model on a grid for capital with 10,000 points uniformly distributed in  $[0.001k_{ss}, 2k_{ss}]$  where  $k_{ss}$  is the steady state value of capital computed in closed form as in chapter 1.

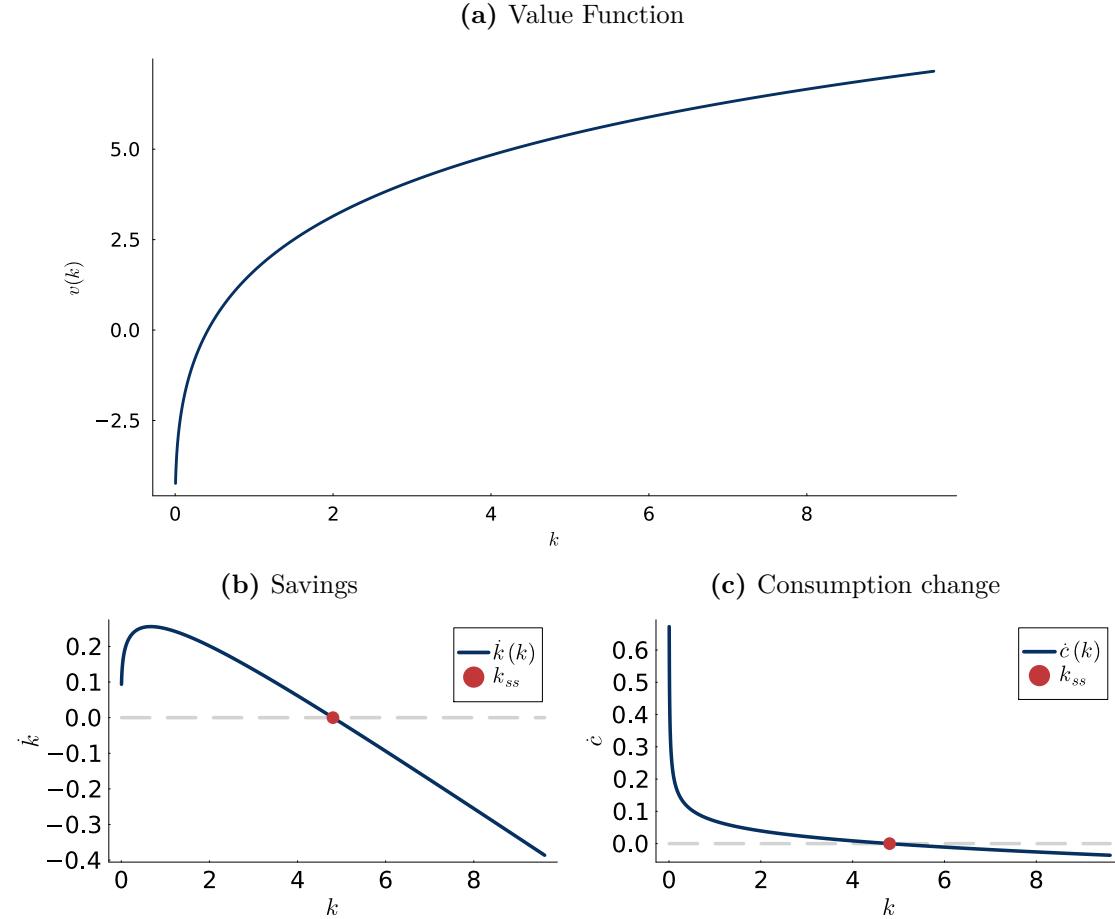


Figure 2.5: Results for the Neoclassical Growth Model

Section B.4 includes the code that was used to generate all results in this chapter. The code is written using the **julia** programming language and its main objective is clarity rather than performance. As a consequence it is certainly not optimized for speed (and is in fact very “unjulian”). Nonetheless, after the first compilation run, solving the model with 10,000 grid points takes less than a second on a standard laptop.

Figure 2.5 plots the approximated value function in panel (a), which is the solution to eq. (2.7), as well as savings  $\dot{k}$  in panel (b) and  $\dot{c}$  in panel (c), which is derived from the Euler equation (2.6). The red dot is the computed value for  $k_{ss}$ . As we can see both  $\dot{k}$  and  $\dot{c}$  cross zero exactly at  $k_{ss}$ , which indicates that the numerical solution is equivalent to the analytical one.<sup>26</sup>

---

<sup>26</sup>Notice that nowhere in the HJB solution did we use the solution for  $k_{ss}$ , except in the definition of the grid (which could clearly have been done without).

(a) Matlab code

```

1 % Elements of the A matrix
2 elem_a = -min(s_B, 0)/dk; % Lower diagonal
3 elem_x = max(s_F, 0)/dk; % upper diagonal
4 elem_b = - elem_a - elem_x; % main diagonal
5
6 % Finite-difference A matrix
7 A = spdiags(elem_a(2:I), -1, I, I) + spdiags(elem_b, 0, I, I) + spdiags([0; elem_x(1:I-1)], 1, I,
   ↵ I);

```

(b) Julia code

```

1 # Elements of the A matrix
2 elem_α = -min.(s_B, 0)/dk # Lower diagonal
3 elem_ξ = max.(s_F, 0)/dk # upper diagonal
4 elem_β = - elem_α - elem_ξ # main diagonal
5
6 # Finite-difference A matrix
7 A = spdiagm(-1 => elem_α[2:end],           # Lower diagonal
8             0 => elem_β[:,],                  # main diagonal
9             1 => elem_ξ[1:(end - 1)]) # upper diagonal

```

Figure 2.6: Examples of how to build a sparse matrix

## 2.3. Some Practical Considerations

Now that we have a basic understanding of how continuous-time methods work, it is worth taking a short detour and answer the question of why we should care about continuous time in the first place. There are at least two main advantages that surfaced in what we covered so far (and more will be apparent in future lectures): First, it is generally somewhat easier to get analytical and numerical results in continuous time due to the fact that first-order conditions are “static”. In fact, just as in the neoclassical growth model we just solved, the FOC  $u'(c) = v'(k)$  relates *current* marginal utility of consumption with the *current* marginal value of capital. On the other hand, in discrete time that same FOC would have gotten us a relation between *current* marginal utility of consumption and *future* value of capital. The intertemporal conditions we get in discrete time usually creates quite a few more problems than the intratemporal ones we get in continuous time.

The second main advantage is that, because time is continuous, variables can only move an infinitesimal amount, which means that transition matrices are extremely sparse. Such sparsity is precisely what allows the numerical solution of continuous-time models to be much faster. In fact, it is *fundamental* to leverage such sparsity in order to take advantage of the computational gains. Specifically, in most common programming languages, it is often necessary to explicitly construct matrices as sparse. By doing so, the program can then specialize the backslash operator to solve the system of equations extremely fast. Figure 2.6 provides two examples of how to explicitly construct a (tridiagonal) sparse matrix in both **Matlab** and **julia**.

As we will see in the coming lectures, continuous time also has an obvious application

### *2.3. Some Practical Considerations*

---

to heterogeneous-agents models because it naturally exploits the connection between the household problem and the problem of finding the distribution – possibly *the* object of interest of heterogeneous-agent models.



# Chapter 3

## Stochastic Calculus

In this lecture we will cover some basic notions of stochastic calculus, focusing mostly on what is necessary to solve heterogeneous-agent models in continuous time. The main reason we are interested in learning about stochastic calculus is that, just as we do in discrete time, we often have to deal with functions of stochastic processes (e.g. expected values). However, in continuous time we cannot apply the standard calculus rules because the Brownian motion – one of the standard building blocks of uncertainty in continuous time – is nowhere differentiable (although everywhere continuous). As a consequence, in order to deal with stochastic processes in continuous time, we will have to adapt some of the rules of standard calculus.

We will start by covering some basic preliminary notions and introducing the main stochastic processes we will use in the rest of these notes. Then, we will present possibly the most important result in stochastic calculus: Itô’s lemma. We will also introduce the Kolmogorov Forward Equation (KFE) – an equation that allows us to characterize the evolution of the probability density function of stochastic processes – and infinitesimal generators, which will later allow us to derive a connection between HJB and KF equations. Finally, we will take a look at how to find a numerical solution to the KFE in order to find the stationary distribution of a specific process.

The objective of these notes is *not* to give an in-depth treatment of stochastic calculus. However, in order to apply many of the methods we are going to use, one needs to be familiar at least with its the basic notions. These notes therefore try to strike a balance of theory and applications. Students who are more interested in directly applying these methods can therefore focus more on the applications, while students who are interested in understanding how and why these methods work will at least find some preliminary notions as well as references to know more. Obviously enough, this means that these notes are in no way a substitute for a book (or a class) on stochastic calculus.

If you are interested in knowing more about stochastic calculus, some useful general references are Øksendal (2014), Karatzas and Shreve (1991) and Karatzas and Shreve (1998), Klebaner (2012), and Kloeden and Platen (1995) (which focuses more on numerical methods for the solution of stochastic differential equations). Other resources that are more economics-focused and you might find useful are Stokey (2009) and Munk (2015)

(the latter is mostly about finance, but with some basic notions of stochastic calculus as well).

## 3.1. Stochastic Processes

Since the objective of stochastic calculus is to work with stochastic processes, it is a good idea to first define what these are.

**Definition 1** [Stochastic Process]: *A process that can be described by the change of some random variables over time.*

**Definition 2** [Stationary Increments]: *For any  $0 < s, t < \infty$ , the distribution of the increments  $W_{t+s} - W_s$  is the same as that of  $W_t - W_0 =: W_t$ .*

**Definition 3** [Independent Increments]: *For every choice of non-negative real numbers  $0 \leq s_1 < t_1 \leq s_2 < t_2 \leq \dots \leq s_n < t_n < \infty$ , the increment random variables  $W_{t_1} - W_{s_1}, W_{t_2} - W_{s_2}, \dots, W_{t_n} - W_{s_n}$  are jointly independent.*

Before defining the various types of stochastic processes, a few preliminaries:

- ▶ We will sometimes refer to continuous processes and discrete processes. However, this does not refer to the time dimension: time is *always* assumed to be continuous. Hence, when we will talk about continuous processes (as opposed to discrete ones) we mean stochastic processes that have continuous sample-paths. That is, processes in which there are no jumps. Very informally, you can think of a continuous process as one which you could in principle draw with a single stroke of a pen (provided you have sufficient ink).
- ▶ Loosely speaking, when we talk about Markov processes, we mean that the evolution of the process can only depend on the history of the process through the *current* value of the process.<sup>1</sup>
- ▶ Throughout this document, we will use  $X_t$  as shorthand notation for  $X(t)$ .

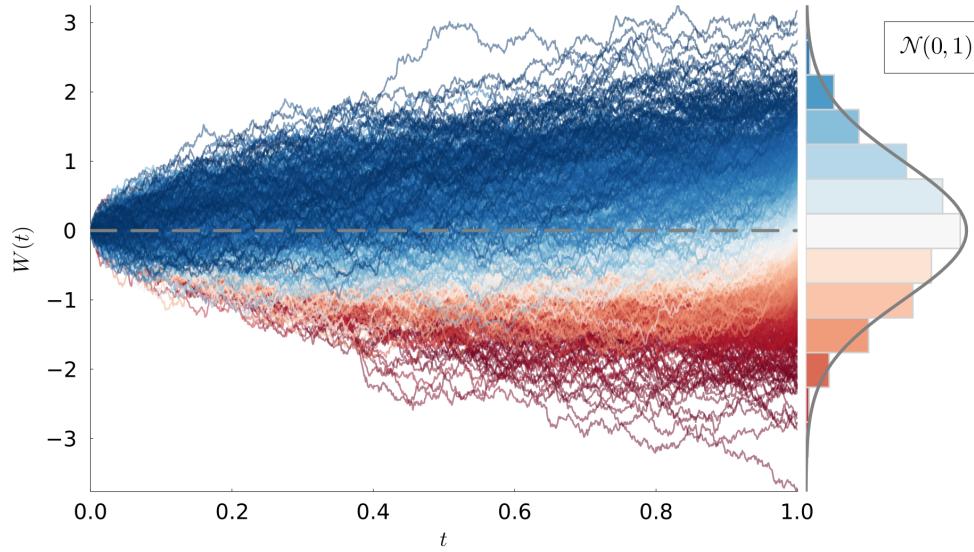
### 3.1.1. Continuous Processes

#### Brownian Motion

In many of the applications we consider in this class, uncertainty is modeled by the evolution of a so-called **standard Brownian motion** (or *Wiener process*), one of the basic building blocks on which we will build more complicated stochastic processes. We will usually think of a Brownian motion  $dW_t$  as the underlying source of uncertainty to the economy or the individual at time  $t$ .

---

<sup>1</sup>We will provide a more precise definition of this property for continuous-time Markov chains in section 3.1.3, while its formal definition for diffusion processes is beyond the scope of these notes and can be found in Øksendal (2014, Theorem 7.1.2.).



**Figure 3.1:** Sample paths of a standard Brownian motion process

**Definition 4** [Standard Brownian Motion]: *A Wiener process is a stochastic process  $W \equiv \{W_t\}_{t \geq 0}$  with the following properties:*

1.  $W_0 = 0$
2.  $W$  has continuous sample paths (the function  $t \rightarrow W_t$  is continuous in  $t$ )
3.  $W$  has stationary, independent increments
4. For all  $0 \leq t_1 < t_2$ , the random variable  $W_{t_2} - W_{t_1}$  is distributed  $\mathcal{N}(0, t_2 - t_1)$ .

Informally, we can write  $dW_t \approx W_{t+dt} - W_t \sim \mathcal{N}(0, dt)$ , which allows us to map the continuous-time process with its discrete-time counterpart using the following approximation:  $W(t + \Delta t) - W(t) = \varepsilon \sqrt{\Delta t}$ , where  $\varepsilon \sim \mathcal{N}(0, 1)$ . In fact, by properties 1 and 4 it is easy to observe that  $W_t \sim \mathcal{N}(0, t)$ . In other words, a Wiener process is essentially the continuous-time analogue of a random walk,  $W_{t+1} = W_t + \varepsilon_{t+1}$  with initial value 0, which also implies that its variance increases with time. While somewhat informal, this approximation will turn out to be particularly valuable when we will need to discretize the process.

Figure 3.1 plots the evolution of a large number of sample paths of a standard Brownian motion, which allows us to directly observe that, as we just argued: the average of the process is zero  $\mathbb{E}(W(t)) = 0$ , the variance increases with time  $\text{Var}(W(t)) = t$ , and the distribution is normal.

**Properties of Brownian motions —** The standard Brownian motion is a martingale, it exhibits the Markov property, it has infinite total variation (over any interval, no matter

how small) and finite quadratic variation.<sup>2</sup> Specifically:

$$\begin{aligned}\langle W, W \rangle_t^1(\omega) &= \infty & \forall t > 0 \\ [W, W]_t &= \langle W, W \rangle_t^2(\omega) = t & \forall t > 0\end{aligned}$$

Intuitively, a process with infinite variation is such that, if we want to draw the sample path over a time interval  $t \in [0, \Delta]$ , no matter how small  $\Delta$  gets, we will always need an infinite amount of ink to draw the process.<sup>3</sup>

Additionally, Brownian motion paths are a continuous function of  $t$  (by definition), are not monotone (in any interval, no matter how small), and are not differentiable at any point.

### Generalized Brownian Motion

By itself, the standard Brownian motion is a rather simple – and therefore not very useful – process. However, as we are about to see, it can be used as a building block for more complex stochastic processes.

For example, a standard Wiener process can be easily generalized to allow for a different drift and volatility. It is in fact often the case that a stochastic process  $\{X_t\}_{t \geq 0}$  is defined in terms of a Brownian motion, an initial value  $X_0$ , and an equation of the form

$$dX_t = \mu_t dt + \sigma_t dW_t \tag{3.1}$$

where  $\mu_t$  and  $\sigma_t$  are known at time  $t$ . For reference, eq. (3.1) is a **stochastic differential equation** (SDE) – a differential equation in which one or more terms (in this case  $dW_t$ ) are stochastic processes.

Taking a step back, let  $W \equiv \{W_t\}_{t \geq 0}$  be a standard Brownian motion, and define a new stochastic process  $X \equiv \{X_t\}_{t \geq 0}$  as

$$X_t = X_0 + \mu t + \sigma W_t, \quad t \geq 0$$

where  $X_0$ ,  $\mu$ , and  $\sigma$  are constants. The change in value of the process between  $t_1$  and  $t_2$ , is then given by:

$$X_{t_2} - X_{t_1} = \mu \cdot (t_2 - t_1) + \sigma \cdot (W_{t_2} - W_{t_1}).$$

Just as before, we can then informally think of  $dX_t$  as the increment  $X_{t+dt} - X_t$  over an “instant” (of length  $dt$ ) and write the change over such infinitesimally short interval  $[t, t + dt]$  (for  $dt \rightarrow 0$ ) in differential form:

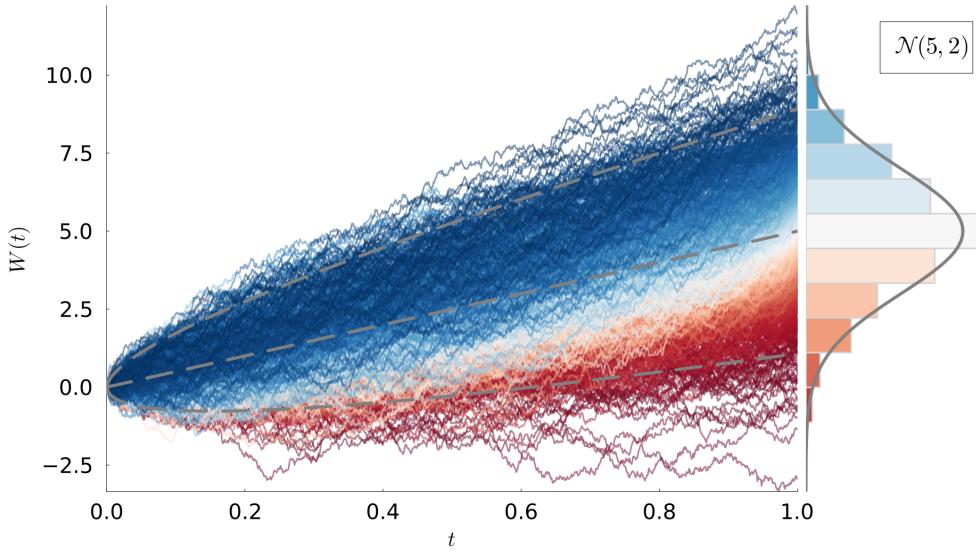
$$dX_t = \mu dt + \sigma dW_t \tag{3.2}$$

Since  $dW_t$  has mean zero and variance  $dt$ , we can also (informally) compute the con-

---

<sup>2</sup>A more complete definition of quadratic variation can be found in section C.2.

<sup>3</sup>In fact, even if we were to zoom in on a sample path of a Brownian motion and we kept zooming on an infinitely small  $dt$  interval, the process would keep looking exactly the same.



**Figure 3.2:** Sample paths of a generalized Brownian motion process

ditional mean and variance of  $dX_t$  as

$$\mathbb{E}_t [dX_t] = \mu dt, \quad \text{Var}_t [dX_t] = \sigma^2 dt$$

and we can interpret  $\mu$  and  $\sigma^2$  as the conditional mean and variance of the change in the value of the process per unit time.

The process  $X$  defined as in eq. (3.2) is called a **generalized Brownian motion** and it is essentially the continuous-time analogue of a random walk with drift. The parameter  $\mu$  is usually referred to as the *drift* of the process, while the parameter  $\sigma$  reflects uncertainty about the future values of the process and is often referred as the *volatility* of the process.

Figure 3.2 plots the evolution of a large number of sample path of a generalized Brownian motion process with  $\mu = 5$  and  $\sigma = 2$ . From this figure we can observe that the process has a clearly positive (and linear) drift, and that the distribution at time  $t$  is  $\mathcal{N}(\mu t, \sigma^2 t)$  (the dashed lines above and below the drift represent the 95% confidence interval).

If the parameters  $\mu$  and  $\sigma$  are allowed to depend on time in a deterministic way,  $X$  is called a *time-inhomogeneous* generalized Brownian motion:

$$dX_t = \mu(t)dt + \sigma(t)dW_t \tag{3.3}$$

or, more precisely:

$$X_{t_2} - X_{t_1} = \int_{t_1}^{t_2} \mu(s)ds + \int_{t_1}^{t_2} \sigma(s)dW_s \tag{3.4}$$

where the last integral is a so-called **stochastic integral** (i.e. an integral which is integrated with respect to a stochastic process).

## Diffusion Processes

Because not all processes can be defined using standard and generalized Brownian motions (e.g. if we want to restrict our stochastic process to be non-negative), it is useful to further generalize our definition to **diffusion processes**:

**Definition 5** [Diffusion Process]: *A diffusion process is a stochastic process  $X = \{X_t\}_{t \geq 0}$  for which the drift  $\mu$  and the volatility  $\sigma$  are allowed to depend on the current value of the process:*

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dW_t$$

A diffusion process is therefore a continuous-time Markov processes with continuous sample paths and, if both  $\mu$  and  $\sigma$  are independent of time, the diffusion is said to be *time-homogeneous*.

Just as before, a more rigorous definition of a diffusion process should start by defining the change in the process over an interval  $[t_1, t_2]$  as

$$X_{t_2} - X_{t_1} = \int_{t_1}^{t_2} \mu(X_s, s)ds + \int_{t_1}^{t_2} \sigma(X_s, s)dW_s. \quad (3.5)$$

Notice that, because the integrands are now functions of  $X_s$  and  $s$ , they are generally unknown at time  $t_1$ . Nonetheless, since both  $\mu(\cdot)$  and  $\sigma(\cdot)$  only depend on the current value of the process, a diffusion process is still a Markov process.

**Ornstein-Uhlenbeck Process** — By choosing appropriate values for  $\mu$  and  $\sigma$ , we can model almost any continuous stochastic process, which demonstrates their remarkable flexibility. One classic example of a diffusion process often used in economics is the Ornstein-Uhlenbeck process (often referred to, especially in finance, as the Vašiček model). In differential form, an Ornstein-Uhlenbeck process can be expressed as:

$$dX = \eta(\bar{X} - X)dt + \sigma dW$$

where  $\bar{X}$  is the long-run mean of the process,  $\sigma$  its instantaneous volatility, and  $\eta$  is the rate of mean reversion. The Ornstein-Uhlenbeck process is the continuous-time analogue of an AR(1) process with autocorrelation  $e^{-\eta} \approx 1 - \eta$ .<sup>4</sup>

Notice also that, if we ignore risk (i.e. set  $\sigma = 0$ ), the Ornstein-Uhlenbeck process reduces to a first-order ODE:

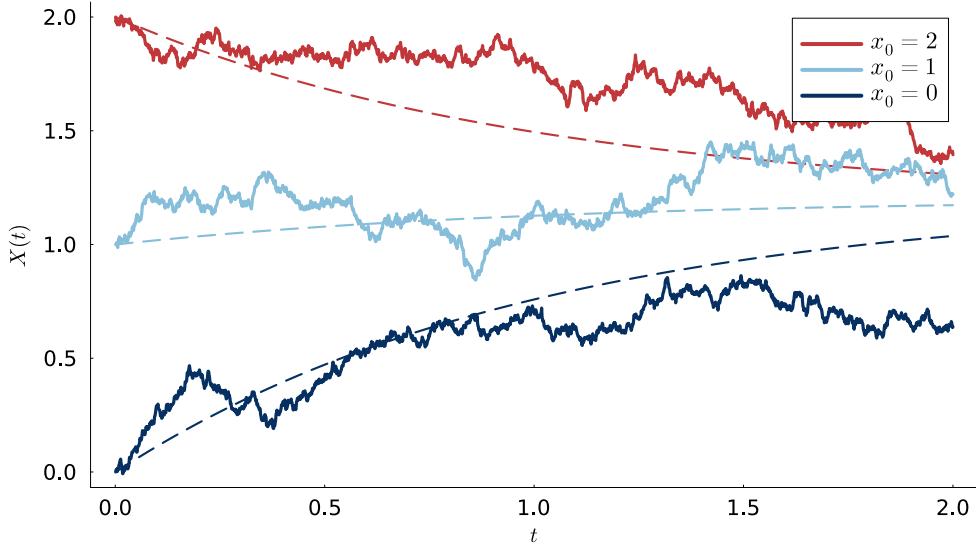
$$\dot{X} = \eta(\bar{X} - X)$$

with solution  $X(t) = \bar{X} + c_0 e^{-\eta t}$  and constant exponential decay  $\eta$  (where  $c_0$  is a constant that will depend on the initial condition  $X(0)$ ).

Figure 3.3 plots the evolution of 3 different sample paths of an Ornstein-Uhlenbeck process with  $\bar{X} = 1.2$ ,  $\eta = 1$ , and  $\sigma = 0.3$ . The paths start from different initial points (0, 1, and 2) and they all tend towards the long-run mean of  $\bar{X}$ . The plot also includes

---

<sup>4</sup>In fact, it can also be shown that the stationary distribution of the process is  $\mathcal{N}\left(\bar{X}, \frac{\sigma^2}{2\eta}\right)$ .



**Figure 3.3:** Three sample paths of an Ornstein-Uhlenbeck process

the paths of the corresponding deterministic ODEs (dashed lines) starting from the same initial points, which shows how each path on average decays towards the mean at rate  $\eta$ .

## Itô Processes

For completeness sake, one can also define continuous processes that are even more general than diffusion processes:

**Definition 6** [Itô Process]: *An Itô process takes the form:*

$$X_t = X_0 + \int_0^t \mu(s)ds + \int_0^t \sigma(s)dW_s \quad (3.6)$$

where  $\mu()$  and  $\sigma()$  are themselves stochastic processes.

The associated SDE in differential form is given by:

$$dX_t = \mu_t dt + \sigma_t dW_t.$$

However notice that, in the case of Itô processes, the more rigorous integral representation of the process in eq. (3.6) is usually more appropriate.

It is easy to see that a diffusion process is simply a special case of an Itô process in which the drift and volatility are simply functions of  $t$  and  $X_t$ . For more general Itô processes, on the other hand,  $\mu_t$  and  $\sigma_t$  may depend for example on past values of  $X$  or of other processes, which implies Itô processes need not be Markov.

In these lectures, we will never actually use Itô processes. However, since some version of theorems – most notably Itô’s lemma – found in the wild often refer to Itô processes, it is useful to at least know what they are. In particular, since a diffusion is just a special

case of an Itô process, any such theorem that applies to the latter will also be applicable to the former.<sup>5</sup>

## Numerical Solution

To conclude this section it should be noted that, except in few and very specific cases, SDEs have no analytical solutions and it is thus usually necessary to approximate them. Researchers have therefore developed plenty of methods to approximate these solutions.<sup>6</sup> The natural first example to discretize an SDE is the **Euler-Maruyama** method, a straightforward extension of the Euler method.

**Euler-Maruyama Method** — Given the following SDE:

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dW_t$$

discretize it along the time dimension to obtain

$$X_{t+\Delta t} - X_t = \mu(X_t, t)\Delta t + \sigma(X_t, t)\Delta W_t.$$

Recalling definition 4, for small  $\Delta t$  we have that  $\Delta W_t \sim \mathcal{N}(0, \Delta t)$  and we can therefore approximate (and simulate)  $\Delta W_t = \varepsilon\sqrt{\Delta t}$  where  $\varepsilon \sim \mathcal{N}(0, 1)$ . Once we have chosen a sufficiently small step-size  $\Delta t$ , and given an initial condition on  $X_0$ , the problem can be solved forward to get a discretized approximation to one (or more) sample path of the underlying continuous-time process. While the Euler-Maruyama method is often inefficient and does not always work well in more complex cases, its simplicity makes it an obvious candidate for simple applications.<sup>7</sup>

### 3.1.2. Discrete Processes

Until now, we have only discussed continuous processes (i.e. without jumps). However, since not all stochastic processes are continuous, it is useful to cover **jump processes** as well.

**Definition 7** [Jump Process]: *A jump process is a stochastic process that has discrete movements (jumps) with random arrival times.*

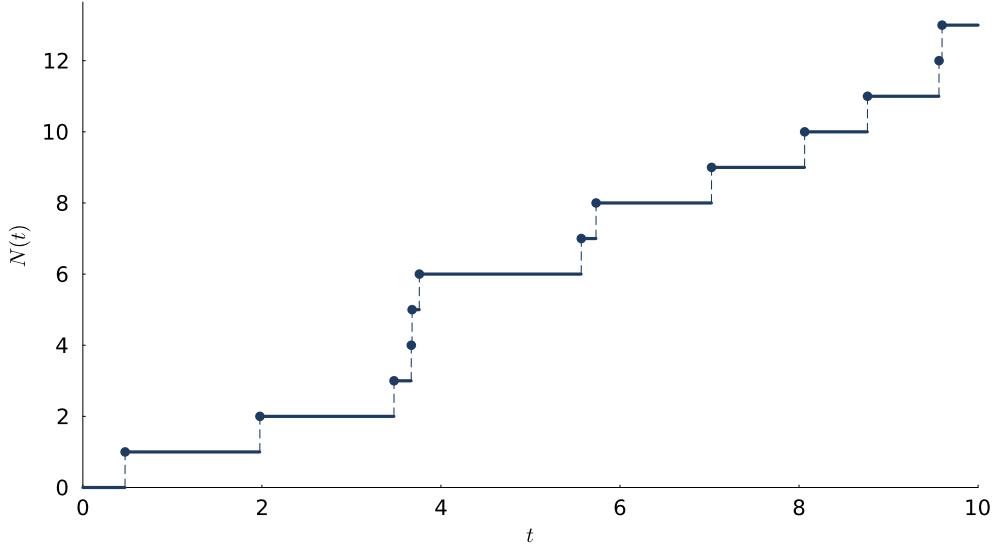
Informally, a jump process is simply a stochastic process that has discontinuities happening at random times.

---

<sup>5</sup> As a side note, the fact that Itô's lemma is often referred to as a theorem is not actually wrong: in fact the reason why it is referred to as a “lemma” is simply historical. As noted in Jarrow and Protter (2004) “The book by H. P. McKean, Jr., published in 1969, had a great influence in popularizing the Itô integral, as it was the first explanation of Itô’s and others’ related work in book form. But McKean referred to Itô’s formula as Itô’s lemma, a nomenclature that has persisted in some circles to this day. Obviously this key theorem of Itô is much more important than the status the lowly nomenclature “lemma” affords it [...]”

<sup>6</sup>Kloeden and Platen 1995, is an excellent resource for many of the “classic” methods which covers everything you might want to know at this stage and more.

<sup>7</sup>In fact, all the figures shown above have been obtained using this method.



**Figure 3.4:** Sample path of a simple Poisson process

The most common type of jump process is the **Poisson process**, which is a type of counting process (i.e. an integer-valued, non-negative, non-decreasing stochastic process which “counts” the number of events that occur in a given time interval).

### Poisson Processes

**Definition 8** [Simple Poisson Process]: *A simple Poisson process (with intensity  $\lambda > 0$ ) is a counting process  $\{N_t\}_{t \geq 0}$  with the following properties:*

1.  $N_0 = 0$
2. *Independent increments*
3. *Stationary Poisson increments: if  $s < t$ , then  $N_t - N_s \sim \text{Pn}(\lambda(t-s))$ :*

$$\mathbb{P}(N_t - N_s = k) = e^{-\lambda(t-s)} \frac{(\lambda(t-s))^k}{k!}, \quad k \geq 0$$

4. *The paths of  $N_t$  are right-continuous with left limits*

In a Poisson process, the intensity  $\lambda$  is the expected number of jumps per unit time, that is:

$$\lambda = \lim_{h \rightarrow 0} \frac{1}{h} \mathbb{P}(N_h = 1)$$

Notice that, even though the number of jumps are distributed as a Poisson distribution (which is a discrete probability distribution), the process itself as a function of time is defined on the real line – which obviously makes it a continuous-time stochastic process.

Figure 3.4 plots one sample path of a simple Poisson process with intensity  $\lambda = 1$ . We can see that the process starts at 0 and increases by a single integer amount at random

times.

The simple Poisson process is by itself rather limited in its applications. Hence, we will often use more general types of jump processes which build on it. Examples of such processes are the compensated Poisson process, which transforms the simple Poisson process into a martingale and is given by  $N_t - \lambda t$ ; the compound Poisson process, which features random jump sizes (e.g. normally distributed); and other extensions including processes with time and/or state-dependent intensities, random intensities, etc.

**Properties of Poisson processes** — A Poisson process is a Markov process with finite total variation and finite quadratic variation.<sup>8</sup> In particular, they are both Poisson processes themselves:

$$\begin{aligned}\langle N, N \rangle_t^1(\omega) &= N_t, & \forall t > 0 \\ [N, N]_t &= \langle N, N \rangle_t^2(\omega) = N_t, & \forall t > 0\end{aligned}$$

Additionally, since the Poisson process has finite variation, we also have that:

$$[W, N]_t = \langle W, N \rangle_t^2(\omega) = 0$$

It can be shown that the probability of observing  $k$  jumps in an interval of length  $h$  is given by:

$$\begin{aligned}\mathbb{P}(N_h = 0) &= e^{-\lambda h} = 1 - \lambda h + o(h) \\ \mathbb{P}(N_h = 1) &= \lambda h e^{-\lambda h} = \lambda h + o(h) \\ \mathbb{P}(N_h = 2) &= h^2 \frac{\lambda^2}{2} = o(h)\end{aligned}$$

where  $o(h) \rightarrow 0$  as  $h \rightarrow 0$ .<sup>9</sup> Therefore, for small  $h$ ,  $N_{t+h} - N_t$  behaves like a Bernoulli random variable with parameter  $\lambda h$ . That is, a jump happens with probability  $\lambda h$  and does not happen with probability  $1 - \lambda h$ .

The time between jumps is exponentially distributed with parameter  $\lambda$  (i.e. it has mean  $1/\lambda$ ).

### 3.1.3. Continuous-time Markov Chains

Before moving on to stochastic calculus, it is worth it to also discuss **continuous-time Markov chains** (CTMC) both because these will be often used in our models and because students who have worked in discrete time will already be familiar with discrete-time

---

<sup>8</sup>A process has finite variation if its paths (realizations of the process over time) have bounded total variation over any finite interval. The paths of a Poisson process are piecewise constant with jumps of size 1 at each event. Since these jumps are countable and each jump is of fixed size, the total variation is simply the number of jumps in the interval, which is finite for any finite interval.

<sup>9</sup> $o(h)$  is the so-called “little-o notation” and it represents a term that goes to zero faster than  $h$ . The first property then simply restates the well known approximation  $e^{\lambda h} \approx 1 + \lambda h$  for small  $h$  (so we are simply rewriting this approximation in a more formal way). The second and third properties come from the fact that terms in  $(\lambda h)^2$  go to zero faster than  $h$  (and therefore go into  $o(h)$ ).

### 3.1. Stochastic Processes

---

Markov chains and with the associated transition matrices. Additionally, since there is a very tight link between transition matrices and infinitesimal generators, treating CTMC separately from other discrete processes will help us gain some intuition about what infinitesimal generators are and how they work in a more familiar context.

In discrete time, a Markov chain remains in any given state for exactly one unit of time before making a transition (possibly to the same state). In continuous time, this assumption is instead relaxed and the chain can remain in any given state for an unspecified amount of time, called *holding time*. While at first sight this might seem to be an issue for the Markov property we are just about to see that, as long as the holding times are exponentially distributed, such property is in fact retained.

In order to define a CTMC, let's first give an appropriate definition of the Markov property in continuous time. Specifically, for a process taking values in a discrete set of states the Markov property requires that:

$$\mathbb{P}(X_{t_0+t} = j | X_{t_0} = i, X_{t'} = i') = \mathbb{P}(X_{t_0+t} = j | X_{t_0} = i) = P_{i,j}(t) \quad \forall 0 \leq t' < t_0. \quad (3.7)$$

That is to say: the probability that the process at a time  $t_0 + t$  is in state  $j$  is only a function of where the process is at time  $t_0$  and of the time interval considered, and does not therefore depend on where the process was at any time  $t' < t_0$ . Importantly, note that this probability  $P_{i,j}(t)$  is (usually) a function of time.

At this point, we are ready to define a continuous-time Markov chain:

**Definition 9** [Continuous-time Markov chain]: *A stochastic process  $X$  taking values in a discrete set of states  $i \in \mathcal{X}$  is called a continuous-time Markov chain if for all  $t_0 \geq 0, t \geq 0, i \in \mathcal{X}, j \in \mathcal{X}$ ,  $X(t)$  satisfies the Markov property.*<sup>10</sup>

That is, “a continuous-time Markov chains is a stochastic process having the Markovian property that the conditional distribution of the future  $X(t+s)$  given the present  $X(s)$  and the past  $X(u), 0 \leq u < s$ , depends only on the present and is independent of the past.” (Ross 2014, p. 372) Informally, a CTMC is nothing else than a continuous-time Markov process taking values in a discrete set of states.

To see why this definition must imply that the holding times are exponentially distributed, suppose that a CTMC enters some state  $i$  at time 0 and suppose that the process does not leave state  $i$  by time  $s$ ; by definition, the Markovian property requires that the probability that the process will remain in state  $i$  for some additional time  $t$  must not depend on how much time it has been in state  $i$  before (i.e. it must not depend on  $s$ ). As a consequence the holding time, which is itself a random variable, is *memoryless* and must therefore be exponentially distributed.<sup>11</sup>

Concluding, “a continuous-time Markov chains is a stochastic process that moves from state to state in accordance with a (discrete-time) Markov chain, but it is such that the

---

<sup>10</sup>We actually also require  $X(t)$  to be right-continuous. However, for most practical purposes, this assumption is often not necessary.

<sup>11</sup>A well-known result in probability theory is that a random variable is memoryless if and only if it is exponentially distributed (See Ross 2014, sec. 5.2.2).

amount of time it spends in each state, before proceeding to the next state, is exponentially distributed. In addition, the amount of time the process spends in state  $i$ , and the next state visited, must be independent random variables” (Ross 2014, p. 373).<sup>12</sup>

**Properties of CTMC** — With our definition on hand, we can also define the transition probability matrix within time  $t$  as:  $\mathbf{P}(t) = [P_{i,j}(t)]$ , i.e. the collection of the transition probabilities from all states  $i \in \mathcal{X}$  to all states  $j \in \mathcal{X}$  in an interval of length  $t$ .

The fact that the matrix  $\mathbf{P}(t)$  is parameterized by time means that we can also apply notions from calculus (because time is continuous). For example, we have that:

$$\mathbf{P}(0) \equiv \lim_{t \downarrow 0} \mathbf{P}(t) = \mathbf{I}.$$

To understand what this limit implies think about what the  $\mathbf{P}(t)$  matrix is: it represents the matrix of transition probabilities within time  $t$ , so if we set  $t = 0$ , it implies no time at all has passed and the process will have experienced no transitions. The probabilities on the main diagonal of  $\mathbf{P}$ ,  $P_{i,i}$  will therefore all equal 1, and all other elements will equal 0. Hence  $\mathbf{P}(0)$  will equal the identity matrix.

In the same vein as we defined  $\mathbf{P}(t)$  we can also define the state distribution as:  $\boldsymbol{\pi}(t) = [\pi_i(t)]' = [\mathbb{P}(X(t) = i)]'$ , which is nothing else than the (row) vector of probabilities that the process  $X(t)$  is in state  $i$  at time  $t$ . It immediately follows that:

$$\pi_j(t + \tau) = \sum_i \pi_i(t) P_{i,j}(\tau) \tag{3.8}$$

$$\boldsymbol{\pi}(t + \tau) = \boldsymbol{\pi}(t) \mathbf{P}(\tau). \tag{3.9}$$

Hence, the state distribution at time  $t + \tau$  is equal to the distribution at time  $t$  times the transition probability matrix within time  $\tau$ .

A CTMC is therefore defined by two components: a so-called **jump chain** (or the *embedded Markov chain*) that gives us the transition probabilities  $P_{i,j}$ ; and the **holding-time parameters**  $\lambda_i$  that control the amount of time spent in each state.

Finally, from eq. (3.9) one can also easily prove the so-called semigroup property (which is the natural analogue of the Chapman-Kolmogorov equations for discrete-time chains):

$$\mathbf{P}(t+s) = \mathbf{P}(t)\mathbf{P}(s) = \mathbf{P}(s)\mathbf{P}(t) \quad \forall t, s \geq 0. \tag{3.10}$$

## 3.2. Stochastic Calculus

One of the main objectives of stochastic calculus is to be able to take derivatives of stochastic objects. However, because the Brownian motion is nowhere differentiable, our usual methods of differentiation does not work anymore. Our “new” method of differentiation will essentially boil down to being a combination of standard calculus plus some simple

---

<sup>12</sup>Notice that, by our definition of a CTMC, it immediately follows that Poisson processes are CTMC on the state space  $\mathbb{Z}_+$  that always move from state  $i$  to state  $i + 1$  (i.e.  $P_{i,i+1} = 1$ ).

	$dt$	$dW_t$	$dN_t$
$dt$	0	0	0
$dW_t$	0	$dt$	0
$dN_t$	0	0	$dN_t$

**Table 3.1:** Algebraic rules for stochastic calculus

rules. We are now going to (informally) derive some of these rules; specifically, those regarding Brownian motions. However, note that such derivations cannot be directly applied to, for example, Poisson processes. Table 3.1 collects the main rules we care about.

### 3.2.1. Differential Algebra

In order to prove Itô's lemma in section 3.2.2 we will make use of three rules, which are:

$$(dt)^2 = 0, \quad dt \cdot dW = 0, \quad (dW)^2 = dt.$$

To see where these rules are coming from, note the following:

- ▶  $(dt)^2 = 0$  – to see why this is the case, recall that, in continuous time,  $dt$  is an infinitesimal time interval  $dt \rightarrow 0$ . As a consequence  $(dt)^2$  is going to be far smaller than  $dt$  itself (essentially 0) and can therefore be safely ignored.
- ▶  $dt \cdot dW = 0$  – given the definition of a Wiener process, we know that  $dW \sim \mathcal{N}(0, dt)$  so in expectations  $dW$  is zero; therefore we also have that  $\mathbb{E}[dt \cdot dW] = dt \cdot \mathbb{E}[dW] = 0$ . However,  $dt \cdot dW$  being zero in expectations is not sufficient to say that  $dt \cdot dW = 0$ . In order to do so, we need to show that it is *always* zero, which we can do by showing that its variance is also zero. In fact, we have that  $\text{Var}[dt \cdot dW] = (dt)^2 \text{Var}[dW] = (dt)^3$  which is again much smaller than  $dt$  itself and can therefore be safely ignored.
- ▶  $(dW)^2 = dt$  – we use the same approach as before: we know that  $\mathbb{E}[(dW)^2] = \text{Var}[dW] - \mathbb{E}[dW]^2 = dt$  so we know that  $(dW)^2$  is on average equal to  $dt$ . However, we need to show that it is always equal to  $dt$ . In order to do so, we can again show that its variance is zero:

$$\text{Var}[(dW)^2] = \underbrace{\mathbb{E}[(dW)^4]}_{3(dt)^2} - \underbrace{\mathbb{E}[(dW)^2]^2}_{(dt)^2} = 2(dt)^2$$

which means that, again, as  $dt \rightarrow 0$ , its variance goes to zero and  $(dW)^2$  can be approximated by its mean  $dt$ .

With these rules on hand, we are now ready to prove one of the main results of stochastic calculus: Itô's lemma.

### 3.2.2. Itô's Lemma

Itô's Lemma is an indispensable tool for working with random processes in continuous time. In fact, we often need to describe the behavior of functions of diffusion processes and, in order to describe how they vary over time, we need to resort to the rules of stochastic calculus. Ito's lemma is one of the fundamental results of stochastic calculus and is essentially the analog of the chain rule of standard calculus for stochastic processes.<sup>13</sup>

We start by giving the statement in its integral form which is more general, and then in its differential form which, albeit less precise, is usually enough for most purposes. Then, we will look at an informal proof and discuss some intuition. Finally (and mostly for completeness) we will see its multivariate version in differential form.

**Theorem 3** [Itô's lemma, integral form]: *Suppose that  $f(x)$  is a twice continuously differentiable function, and  $X_t$  is an Itô process with*

$$X_t = X_0 + \int_0^t \mu_s ds + \int_0^t \sigma_s dW_s.$$

*Then,*

$$\begin{aligned} f(X_t) &= f(X_0) + \int_0^t f'(X_s) dX_s + \frac{1}{2} \int_0^t f''(X_s) d[X, X]_s \\ &= f(X_0) + \int_0^t f'(X_s) dX_s + \frac{1}{2} \int_0^t f''(X_s) \sigma_s^2 ds \\ &= f(X_0) + \int_0^t \left( f'(X_s) \mu_s + \frac{1}{2} f''(X_s) \sigma_s^2 \right) ds + \int_0^t f'(X_s) \sigma_s dW_s \end{aligned}$$

**Theorem 4** [Itô's lemma, differential form]: *Suppose that  $X_t$  is an Itô process with*

$$dX_t = \mu_t dt + \sigma_t dW_t$$

*and  $f : \mathbb{R} \rightarrow \mathbb{R}$  is a twice continuously differentiable. Then, the process  $Y$  defined by  $Y_t = f(X_t)$  is itself an Itô process with*

$$dY_t = \left( f'(X_t) \mu_t + \frac{1}{2} f''(X_t) \sigma_t^2 \right) dt + f'(X_t) \sigma_t dW_t.$$

### Proof

The proof of Itô's lemma we are going to present here assumes  $\mu_t$  and  $\sigma_t$  are constant, it uses a second-order Taylor expansion of  $f$  and is a version of those often found in introductory textbooks on stochastic calculus. A more rigorous proof can be found in Øksendal 2014.

---

<sup>13</sup>One of the obvious implications of Itô's lemma is that any (twice differentiable) function of a diffusion process is itself going to be a diffusion process.

Let's start by applying a second-order Taylor expansion of  $f$ :

$$dY = \frac{\partial f(X)}{\partial X} dX + \frac{1}{2} \frac{\partial^2 f(X)}{\partial X^2} dX^2 \quad (3.11)$$

The second term,  $dX^2$  is then:

$$dX^2 = (\mu dt + \sigma dW)^2 = \mu^2 dt^2 + 2\mu\sigma dt dW + \sigma^2 dW^2$$

Now, recalling the rules derived earlier:  $dt^2 = 0$ ,  $dt \cdot dW = 0$ , and  $dW^2 = dt$ , we have:

$$dX^2 = \sigma^2 dt$$

and, substituting  $dX$  and  $dX^2$  in eq. (3.11), we get:

$$\begin{aligned} dY &= \frac{\partial f(X)}{\partial X}(\mu dt + \sigma dW) + \frac{1}{2} \frac{\partial^2 f(X)}{\partial X^2} \sigma^2 dt \\ &= \left( f'(X)\mu + \frac{1}{2} f''(X)\sigma^2 \right) dt + f'(X)\sigma dW. \end{aligned}$$

## Intuition

When applying Itô's Lemma, we encounter an unintuitive feature: a second-order term in the drift coefficient.<sup>14</sup> This second-order term is generally absent in ordinary calculus, and is key to understanding the nature of stochastic processes.

Essentially, the derivation of Itô's lemma revealed us that, in the presence of a Brownian motion, the drift of a function of a stochastic process is affected not just by the deterministic drift term (which is analogous to what we'd see in the deterministic calculus) but also by the stochastic diffusion term. The latter is represented by the second-order term that emerges in Itô's calculus, and is absent in ordinary calculus. That is, it is not sufficient to write down

$$dY = \frac{\partial f(X)}{\partial X} dX$$

as we would have done in standard calculus because the second order term  $dX^2$  is non-negligible: when dealing with Brownian motion processes, both the first and second derivatives matter (which is a direct consequence of the fact that Wiener processes have finite quadratic variation). The second-order term in the Itô's lemma is therefore a direct reflection of the stochastic nature of the process we are dealing with.

In order to develop some intuition for this, it might be useful to turn to Jensen's inequality. Recall that Jensen's inequality states that for a convex function, the expectation of the function of a random variable is greater than or equal to the function of the expectation of the random variable.

In the context of Itô's Lemma, this relationship is critical. In fact, the second-order term in Itô's Lemma that enters the drift is a direct consequence of Jensen's inequality acting on the quadratic variation of the stochastic process. Because a stochastic process

---

<sup>14</sup>We will sometimes refer to such term as Itô- (or variance-) correction term.

has non-zero variance (i.e., it's spread out), the function of the expectation is affected by the shape of the function itself – which is given by its second derivative.

To put it simply, the extra term in Itô's lemma arises because of Jensen's inequality and the stochastic nature of the process: since the variance of a stochastic process is non-zero, the expected value of a function of that process will not simply be the function of the expected value of the process – instead, it will be affected by the “spread” of the process. The second derivative, which measures the curvature of the function, tells us how this spread affects the expected value.

**Geometric Brownian motion** — Consider, the following example: let  $X$  be a so-called geometric Brownian motion (i.e.  $dX = \mu X dt + \sigma X dW$ ) and suppose we want to know what is the evolution of the logarithm of a Brownian motion:  $Y(X) = \log(X)$ . In order to find that out, we simply need to apply Itô's lemma: the first and second derivative of  $Y(X) = \log(X)$  are  $Y'(X) = 1/X$  and  $Y''(X) = -1/X^2$ , respectively. Therefore we obtain:

$$\begin{aligned} dY &= \left( (\mu X) \frac{1}{X} - \frac{1}{2} (\sigma X)^2 \frac{1}{X^2} \right) dt + (\sigma X) \frac{1}{X} dW \\ &= \left( \mu - \frac{\sigma^2}{2} \right) dt + \sigma dW \end{aligned}$$

which is a standard Brownian motion with drift and diffusion coefficients given by  $\mu - \sigma^2/2$  and  $\sigma$ , respectively. As a consequence  $Y$  does will be distributed as:

$$Y(t) \sim \mathcal{N} \left( \left( \mu - \frac{\sigma^2}{2} \right) t, \sigma^2 t \right)$$

that is, the logarithm of a geometric Brownian motion is a Brownian motion.

## Multivariate Itô's Lemma

For completeness' sake, we hereby report the multivariate version of Itô's lemma

**Lemma 1** [Itô's lemma, multivariate]: *Let  $\mathbf{x} \in \mathbb{R}^n$ ,  $\boldsymbol{\mu} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  and  $\boldsymbol{\sigma} : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^n$ . Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , with gradient  $\nabla_{\mathbf{x}} f$  and Hessian  $\nabla_{\mathbf{x}}^2 f$ , then*

$$df(\mathbf{x}) = \left( (\nabla_{\mathbf{x}} f)^T \boldsymbol{\mu} + \frac{1}{2} \text{tr} (\boldsymbol{\sigma}^T (\nabla_{\mathbf{x}}^2 f) \boldsymbol{\sigma}) \right) dt + (\nabla_{\mathbf{x}} f)^T \boldsymbol{\sigma} d\mathbf{W}_t$$

where  $\text{tr}()$  is the trace and  $\mathbf{W}_t$  is an  $m$ -dimensional Brownian motion.<sup>15</sup>

---

<sup>15</sup>The trace of a matrix  $\mathbf{A}$ , denoted  $\text{tr}(\mathbf{A})$ , is defined to be the sum of elements on the main diagonal of  $\mathbf{A}$ . Notice that the trace is only defined for square matrices.

### 3.3. The Kolmogorov Forward Equation

Now that we have introduced stochastic processes, and have (briefly) seen how to generate specific sample paths from them, it is only natural to want to look at how a stochastic process is distributed. That is, if we could generate an infinite number of sample paths from a specific stochastic process, how would the *distribution* of these sample paths look like? and additionally, starting from any initial distribution (e.g. a point mass) how would the distribution evolve over time? All of these questions, and more, can be answered by looking at the so-called **Kolmogorov Forward** equation (KFE), also known as the Fokker-Plack equation. The KFE is nothing else than an equation characterizing the evolution of the probability density function of a stochastic process. As we will see, this equation will be a key tool when solving continuous-time heterogeneous-agent models.

**Proposition 1** [Kolmogorov Forward Equation]: *Let  $X_t$  be an Itô process described by:*

$$dX = \mu(X, t)dt + \sigma(X, t)dW.$$

*Then,  $g(x, t)$  – the distribution of  $X$  at time  $t$  – satisfies the following Kolmogorov Forward equation.<sup>16</sup>*

$$\frac{\partial g(x, t)}{\partial t} = -\frac{\partial}{\partial x} [\mu(x, t)g(x, t)] + \frac{1}{2} \frac{\partial^2}{\partial x^2} [\sigma^2(x, t)g(x, t)]. \quad (3.12)$$

The KFE is a **partial differential equation** (PDE) – a differential equation involving a function of two or more variables and its partial derivatives with respect to those variables (i.e. a differential equation that contain partial derivatives) – which defines how the “cross-sectional” distribution of the process,  $g$ , evolves over time.<sup>17</sup>

A consequence of eq. (3.12) is that if a stationary distribution of the process,  $g(x) \equiv \lim_{t \rightarrow \infty} g(x, t)$ , exists it solves:

$$0 = -\frac{\partial}{\partial x} [\mu(x)g(x)] + \frac{1}{2} \frac{\partial^2}{\partial x^2} [\sigma^2(x)g(x)]$$

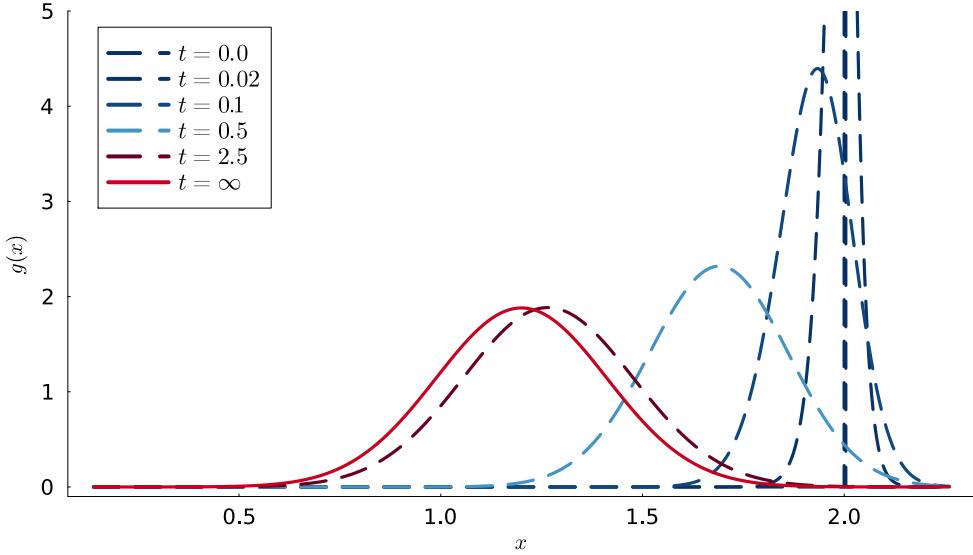
where we have simply used the fact that, because the distribution is not changing over time,  $\frac{\partial g(x, t)}{\partial t} = 0$ . Like for SDEs, PDEs do not generally have closed-form solutions except in very specific cases and so they need to be solved numerically. Since they involve partial derivatives, in order to numerically solve PDEs, we will have to find a way to approximate such derivatives, just as we did when solving the HJB of the neoclassical growth model (which was also, incidentally, a PDE).

As an example, in fig. 3.5, we use the KFE to look at how the distribution of an

---

<sup>16</sup>A derivation of the KFE for univariate diffusion processes can be found in section C.1.

<sup>17</sup>The KFE is not to be confused with the Kolmogorov Backward Equation, which is a different, although linked, PDE: the KFE allows us to know what the probability distribution of the state will be at time  $s > t$  given an initial condition at time  $t$  (the KFE is therefore integrated *forward* in time). On the other hand, the backward equation describes what is the probability at time  $t$  of ending up in a “target set” at time  $s > t$ , and therefore uses a final condition on the PDE (which is then integrated *backward* in time from  $s$  to  $t$ ).



**Figure 3.5:** Evolution of the distribution for an Ornstein-Uhlenbeck process.

Ornstein-Uhlenbeck process evolves over time. Specifically, we use the same process as in fig. 3.3 – namely we set  $\bar{X} = 1.2$ ,  $\eta = 1$ , and  $\sigma = 0.3$  – and simulate its evolution over time. As initial condition we assume that the process starts from a single mass point at  $x_0 = 2$ , that is we use the Dirac delta function centered at 2,  $\delta_2(x)$ . The figure then shows, how the distribution evolves over time until it eventually reaches the stationary distribution – which we know is  $\mathcal{N}(\bar{X}, \frac{\sigma^2}{2\eta})$ .

**Multivariate Case —** The KFE (3.12) can easily be generalized to the multivariate case, which we report here for completeness:

**Proposition 2** [KFE, multivariate]: *Let  $\mathbf{x}_t \in \mathbf{R}^n$  be an Itô process described by:*

$$d\mathbf{x}_t = \boldsymbol{\mu}(\mathbf{x}_t, t)dt + \boldsymbol{\sigma}(\mathbf{x}_t, t)d\mathbf{W}_t$$

where  $\boldsymbol{\mu}$  is an  $n$ -dimensional vector,  $\boldsymbol{\sigma}$  is an  $n \times m$  dimensional matrix and  $\mathbf{W}_t$  is an  $m$ -dimensional vector of Brownian motions. The KFE for the joint distribution  $g(\mathbf{x}, t)$  is:

$$\frac{\partial g(\mathbf{x}, t)}{\partial t} = - \sum_{i=1}^n \frac{\partial}{\partial x_i} [\mu_i(\mathbf{x}, t)g(\mathbf{x}, t)] + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2}{\partial x_i \partial x_j} [(\boldsymbol{\sigma}\boldsymbol{\sigma}')_{i,j}g(\mathbf{x}, t)] \quad (3.13)$$

### 3.4. Infinitesimal Generators

As we will see more in details in the next two lectures, infinitesimal generators are particularly useful when mapping stochastic differential equations to their numerical approximations. However, before defining infinitesimal generator for generic continuous-time

processes, it is useful to look at how they are defined (and how they work) for continuous-time Markov chains as they are tightly linked to the more familiar transition matrices.

### 3.4.1. Generator of a CTMC

Recall we defined the transition probability matrix of a CTMC within time  $t$  as the collection of all transition probabilities from states  $i$  to states  $j$ :  $\mathbf{P}(t) = [P_{i,j}(t)]$ .

The infinitesimal generator of a CTMC represents the rate of change of these transition probabilities over time and is itself a matrix. Essentially, if we define  $\mathbf{Q}$  as the infinitesimal generator of a CTMC, its elements  $q_{i,j}$  are nothing else than the time derivative  $P'_{i,j}(t)$  evaluated at  $t = 0$ . More formally, except for pathological cases, the following limit exists and defines the **infinitesimal generator**  $\mathbf{Q}$  of a CTMC:<sup>18</sup>

$$\mathbf{Q} = \mathbf{P}'(0) \equiv \lim_{h \downarrow 0} \frac{\mathbf{P}(h) - \mathbf{I}}{h}. \quad (3.14)$$

Since in the coming lectures when dealing with both CTMC and other processes we will usually work with infinitesimal generators (as opposed to transition probability matrices), and many of the properties of CTMC generators translate one to one to more complex processes it is worth taking a closer look at what they are and how they work.

**Properties of CTMC generators** — First, notice that since all rows of the transition probability matrix  $\mathbf{P}(h)$  must add up to one, it must be the case that the rows of  $\mathbf{Q}$  sum to zero. In order to see why this is the case, we can write:

$$\begin{aligned} \sum_{j \neq i} P_{i,j}(h) &= 1 - P_{i,i}(h) \\ \sum_{j \neq i} \frac{P_{i,j}(h)}{h} &= \frac{1 - P_{i,i}(h)}{h} \\ \lim_{h \downarrow 0} \sum_{j \neq i} \frac{P_{i,j}(h)}{h} &= -\lim_{h \downarrow 0} \frac{P_{i,i}(h) - 1}{h} \\ \sum_{j \neq i} q_{i,j} &= -q_{i,i} \end{aligned}$$

where  $q_{i,j}$  are simply the elements of  $\mathbf{Q} = [q_{i,j}]$ .

Second, since the rows of  $\mathbf{Q}$  must sum up to zero, it must be the case that in any row  $i$  some entries are positive and some are negative. Specifically, it turns out to be the case that all off-diagonal entries are positive and all on-diagonal entries are negative. To see why notice that, as we already said, the elements  $q_{i,j}$  represent the rate of change in the transition probability from  $i$  to  $j$ . Hence, since as time progresses the probability of moving from state  $i$  to state  $j$  goes up, by construction it must be the case that  $q_{i,j} \geq 0 \forall i \neq j$ ; similarly, since time the probability of staying in state  $i$  goes down over time, it must be the case that  $q_{i,i} \leq 0$ . Informally, we can think of  $q_{i,i}$  as the “outflow” from state  $i$  and

---

<sup>18</sup>You should think of the  $\mathbf{I}$  on the right-hand side of eq. (3.14) as  $\mathbf{P}(0)$ .

of  $q_{i,j}$  as the “inflow” from state  $i$  to state  $j$  and that these flows have to net to zero: all flows going out of state  $i$  must go to some other state  $j$ .

What if we wanted to compute the rate of change in the transition probabilities at any other  $t \neq 0$ ? From the Chapman-Kolmogorov eq. (3.10) we have that

$$\begin{aligned}\mathbf{P}(t+h) - \mathbf{P}(t) &= \mathbf{P}(t)\mathbf{P}(h) - \mathbf{P}(t) = \mathbf{P}(t)[\mathbf{P}(h) - \mathbf{I}] \\ &= \mathbf{P}(h)\mathbf{P}(t) - \mathbf{P}(t) = [\mathbf{P}(h) - \mathbf{I}]\mathbf{P}(t).\end{aligned}$$

Dividing both sides by  $h$  and taking the limit as  $h \downarrow 0$  we find:

$$\frac{d}{dt}\mathbf{P}(t) = \mathbf{P}(t)\mathbf{Q} \tag{3.15}$$

$$= \mathbf{Q}\mathbf{P}(t). \tag{3.16}$$

which are nothing else than the Kolmogorov forward and backward equations, respectively.

When coupled with the initial condition  $\mathbf{P}(0) = \mathbf{I}$ , the matrix differential eqs. (3.15) and (3.16) have solution:

$$\mathbf{P}(t) = e^{\mathbf{Q}t} \tag{3.17}$$

which, if we forget for one second that we are dealing with matrices, closely resembles the solution to a first-order ODE of the form  $\dot{p}(t) = p(t)q$ . In fact, it turns out that once we have the generator  $\mathbf{Q}$ , we can directly solve eq. (3.17) to obtain the transition matrix  $\mathbf{P}(t)$  for any duration  $t$  (and viceversa). However, because we are dealing with matrices, one has to be careful to apply matrix exponentiation, as opposed to element-wise exponentiation, since the two are *not* the same.<sup>19</sup>

In order to gain some further intuition about these transition rates, we can also look at how they relate to holding times – the amount of time the chain stays in a given state. Defining as  $T_i$  the holding time in state  $i$ , we saw earlier that  $T_i$  is a random variable that, in order for the process to be Markovian, must be exponentially distributed. Let  $\lambda_i$  be the rate parameter of its exponential distribution, i.e.

$$\mathbb{P}(T_i > t) = e^{-\lambda_i t}, \quad t \geq 0.$$

By the Markov property we have that  $\mathbb{P}(T_i > t + \tau | T_i > \tau) = P_{i,i}(t)$ . If we let  $\tau = 0$ , we have  $\mathbb{P}(T_i > t) = e^{-\lambda_i t} = P_{i,i}(t)$ . Differentiating both sides and evaluating at  $t = 0$ , we get:

$$\begin{aligned}\frac{d}{dt}e^{-\lambda_i t}\Big|_{t=0} &= \frac{d}{dt}P_{i,i}(t)\Big|_{t=0} \\ -\lambda_i e^{-\lambda_i \cdot 0} &= P'_{i,i}(0) \\ \lambda_i &= -q_{i,i}\end{aligned}$$

---

<sup>19</sup>There are several numerical methods to compute matrix exponentiation but most numerical software usually include at least one such method (as long as you call the right function).

where from the second to third line we have simply used the definition of  $\mathbf{Q}$ . Hence,  $T_i \sim \text{Exp}(-q_{i,i})$  and the mean holding time in state  $i$  is simply  $\mathbb{E}[T_i] = -\frac{1}{q_{i,i}}$  (recall that  $q_{i,i} \leq 0$ ). In other words, the elements on the main diagonal of the infinitesimal generator of the CTMC are nothing else than (minus) the rate parameter of the holding times and consequently also the (inverse) average holding times.

Finally, it is useful to look at how the generator of a process relates to the state distribution  $\boldsymbol{\pi}(t)$ . Recall the definition of the state distribution  $\boldsymbol{\pi}(t)$  in eq. (3.8), which we can write as:

$$\begin{aligned}\pi_j(t+h) &= \sum_i \pi_i(t) P_{i,j}(h) \\ \frac{\pi_j(t+h) - \pi_j(t)}{h} &= \pi_j(t) \frac{P_{j,j}(h) - 1}{h} + \sum_{i \neq j} \pi_i(t) \frac{P_{i,j}(h)}{h}\end{aligned}$$

where in the second line we have simply subtracted  $\pi_j(t)$  and divided by  $h$  on both sides. Taking the limit as  $h \downarrow 0$  we have:

$$\frac{d}{dt} \pi_j(t) = \sum_i \pi_i(t) q_{i,j} \quad (3.18)$$

$$\frac{d}{dt} \boldsymbol{\pi}(t) = \boldsymbol{\pi}(t) \mathbf{Q}. \quad (3.19)$$

Equation (3.19) can therefore be used to get the state distribution at any time  $t$  from any initial distribution  $\boldsymbol{\pi}(0)$  simply as  $\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0) e^{\mathbf{Q}t} = \boldsymbol{\pi}(0) \mathbf{P}(t)$ .

Obviously enough, if the steady-state distribution  $\boldsymbol{\pi} = \lim_{t \rightarrow \infty} \boldsymbol{\pi}(t)$  exists, it must be the case that  $\frac{d}{dt} \boldsymbol{\pi} = 0$  and therefore, from eq. (3.19), the steady state distribution must also satisfy the following KFE.<sup>20</sup>

$$\boldsymbol{\pi} \mathbf{Q} = 0$$

which can be rewritten less compactly as

$$0 = \sum_i \pi_i q_{i,j} = \sum_{i \neq j} \pi_i q_{i,j} + \pi_j q_{j,j} = \sum_{i \neq j} \pi_i q_{i,j} + \pi_j \left( -\sum_{i \neq j} q_{i,j} \right). \quad (3.20)$$

This last way of formulating the KFE should also hopefully clarify the interpretation of the  $q_{j,j}$  as outflows from state  $j$  and of the  $q_{i,j}$  as inflows from state  $i$  to state  $j$ .

---

<sup>20</sup>In fact one can also show that, up to a normalization, the steady-state distribution  $\boldsymbol{\pi}$  equals the eigenvector associated to the first eigenvalue of  $\mathbf{Q}$  (which is 0, by construction).

### 3.4.2. Generators of General Processes

**Definition 10** [Infinitesimal Generator]: *The infinitesimal generator,  $\mathcal{A}$ , of a Markov process is a linear operator that captures the evolution of the process over time:*<sup>21</sup>

$$\mathcal{A}f(x) = \lim_{t \downarrow 0} \frac{\mathbb{E}[f(X_t)|X_0 = x] - f(x)}{t} \quad (3.21)$$

The basic idea behind the infinitesimal generator is as follows: knowing that the process  $X_t$  is at  $x$ , how does the evolution of the process affect the value of  $f(X_t)$  in an infinitesimal amount of time? In that sense, we can see that the definition of the infinitesimal generator is perfectly consistent (and is in fact the same) with the generator we derived for CTMC.

In section 3.4.1 we discussed the connection between infinitesimal generators for CTMC and transition matrices; the same connection applies here for more general processes. So, if you find definition 10 somewhat confusing, you can keep thinking about the generator as encoding information about the transition *rate* of the process.

As an example, if we restrict our attention to time-homogeneous diffusion processes

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW$$

and considering a generic function  $f \in C^2$  (twice continuously differentiable), it is a simple application of Itô's lemma to the expected value in eq. (3.21) to show that the infinitesimal generator of the above diffusion is given by:

$$(\mathcal{A}f)(x) = \mu(x)f'(x) + \frac{1}{2}\sigma^2(x)f''(x). \quad (3.22)$$

As we will see in the next lecture there is a very tight link between infinitesimal generators, HJB and KF equations. However, before doing so, it is useful to also define the **Hermitian adjoint** (or simply adjoint) of an operator.

### Adjoint Operators

**Definition 11** [Hermitian Adjoint]: *The Hermitian adjoint of an operator  $\mathcal{A}$  is the operator  $\mathcal{A}^*$  such that:*

$$\langle \mathcal{A}x, y \rangle = \langle x, \mathcal{A}^*y \rangle \quad (3.23)$$

where  $\langle \cdot, \cdot \rangle$  is the inner product on vector spaces: e.g.  $\langle u, h \rangle = \int u(x)h(x)dx$ .<sup>22</sup>

That is,  $\mathcal{A}^*$  must be such that:

$$\langle \mathcal{A}f, g \rangle \equiv \int_{-\infty}^{\infty} \mathcal{A}f(x)g(x)dx = \int_{-\infty}^{\infty} f(x)\mathcal{A}^*g(x)dx \equiv \langle f, \mathcal{A}^*g \rangle.$$

In fact, given an infinitesimal generator  $\mathcal{A}$  one can easily (if somewhat tediously) find the

---

<sup>21</sup>The fact that the infinitesimal generator is a *linear* operator will turn out to be very useful when solving these models numerically.

<sup>22</sup>The inner product is simply a generalization of the dot product to infinite-dimensional spaces.

adjoint  $\mathcal{A}^*$  by simply using integration by parts on the above equation.

For the generator of a time-homogeneous diffusion process as in eq. (3.22), it can be shown that its adjoint  $\mathcal{A}^*$  is given by:

$$(\mathcal{A}^* g)(x) = -\frac{\partial}{\partial x} [\mu(x)g(x)] + \frac{1}{2} \frac{\partial^2}{\partial x^2} [\sigma^2(x)g(x)] \quad (3.24)$$

which the careful reader might have already recognized as the right hand side of the KF eq. (3.12). In fact, it is easy to see that we can rewrite that equation more compactly as:

$$\partial_t g = \mathcal{A}^* g \quad (3.25)$$

where  $\partial_t g = \frac{\partial g}{\partial t}$ .

In order to try and build some intuition about adjoint operators, it is useful to mention that in finite dimensions – where operators can be represented by matrices (e.g. as in the case of CTMC) – the Hermitian adjoint is exactly equivalent to the conjugate transpose. It immediately follows that, if all the elements of a matrix are real – as they usually are in economics applications – the Hermitian adjoint and transpose are the same.<sup>23</sup> In this sense, if the concept of adjoint seem too abstract and unclear, we can simply think of it as the generalization to infinite dimensions of the matrix transpose.

Because when solving our models numerically we will always have to approximate the state space in finite dimensions, this connection between adjoint operators and matrix transposes is going to be extremely useful. In fact, if we can write the finite-dimensional representation of an operator  $\mathcal{A}$  we will also get the equivalent representation of its adjoint  $\mathcal{A}^*$  “for free” (at the cost of one matrix transposition).

## 3.5. Example: Ornstein-Uhlenbeck process

In order to better understand the concepts of infinitesimal generator, adjoint, and Kolmogorov Forward Equation, we can use the now familiar Ornstein-Uhlenbeck process as an example and look at how to find its stationary distribution. By definition the process can be written as:

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW_t$$

where  $\mu(X_t) = \eta(\alpha - X_t)$  and  $\sigma(X_t) = \sigma$ .

Using eq. (3.22), we can directly apply the formula to write down the infinitesimal generator of the process:

$$(\mathcal{A}f)(x) = \eta(\alpha - x)f'(x) + \frac{\sigma^2}{2}f''(x). \quad (3.26)$$

Notice that eq. (3.26) is surprisingly similar to the right hand side of the HJB equation in chapter 2: the first term is given by the drift of the state variable times the first

---

<sup>23</sup>The conjugate transpose of a matrix is the same as the transpose except that along with switching rows and column elements one also need to take the complex conjugate of all elements.

derivative of the function being considered; while the second term is an additional new one due to the presence of the stochastic term  $dW_t$ . In fact, when writing down the finite-difference matrix  $\mathbf{A}$  in chapter 2 what we were doing was exactly finding a finite-difference approximation to the infinitesimal generator  $\mathcal{A}$ .

Alternatively, using eq. (3.24) we can also directly write down the adjoint of  $\mathcal{A}$  instead:

$$(\mathcal{A}^* g)(x) = -\frac{\partial}{\partial x} [\eta(\alpha - x)g(x)] + \frac{1}{2} \frac{\partial^2}{\partial x^2} [g(x)\sigma^2] \quad (3.27)$$

where, as we just argued in the last section, eq. (3.27) is nothing else than the right hand side of the KFE. Hence, in order to find the stationary distribution of our process we will need to find the  $g$  such that eq. (3.27) equals zero. In the specific case of the Ornstein-Uhlenbeck process this could in principle be done analytically. However, since this cannot usually be done for more complicated processes, we will instead solve eq. (3.27) numerically. Specifically, just as we solved the HJB in chapter 2, we will write a finite-difference approximation of  $\mathcal{A}^*$  and then find a way to set the discretized KFE to zero.

### 3.5.1. Numerical Solution

We already saw in chapter 2 how to find a finite-difference approximation of the drift term in  $\mathcal{A}$  by using an upwind scheme and we previously argued that we can get the FD approximation of  $\mathcal{A}^*$  simply as the matrix transpose of (the FD approximation of)  $\mathcal{A}$ . That is, if we could also find a FD approximation of the diffusion term in  $\mathcal{A}$ , we would be one matrix transpose away from finding a FD approximation of  $\mathcal{A}^*$ . Summarizing, in order to numerically approximate  $\mathcal{A}^*$ , we need to do two things: first, define a finite-difference approximation of  $\frac{1}{2}\sigma^2(x)f''(x)$  – which allows us to build our numerical approximation  $\mathbf{A}$  of  $\mathcal{A}$  – then, to find the approximation of  $\mathcal{A}^*$ , we simply need to take the transpose of  $\mathbf{A}$ .<sup>24</sup>

To approximate  $\frac{1}{2}\sigma^2(x)f''(x)$  we obviously need an approximation of the second derivative  $f''$ . It turns out that the following central difference approximation is a good choice:

$$f''(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \approx \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta_i^2} \quad (3.28)$$

where  $f_{i-1}$ ,  $f_i$  and  $f_{i+1}$  are the values of the function  $f$  at the points  $x_{i-1}$ ,  $x_i$  and  $x_{i+1}$   $\forall i \in 2, \dots, I-1$ , respectively, considering a grid of point on which we approximate the function  $\{x_j\}_{j=1}^I$ .<sup>25</sup>

---

<sup>24</sup>We could in principle discretize  $\mathcal{A}^*$  directly. However, because the “time directions” of  $\mathcal{A}$  and  $\mathcal{A}^*$  are opposite of each other, the upwind scheme would then need to be modified accordingly. It is therefore a lot simpler (and in line with what we will do when solving HA models) to first approximate  $\mathcal{A}$  by  $\mathbf{A}$  and then find its adjoint  $\mathcal{A}^*$  by taking the transpose  $\mathbf{A}^T$ .

<sup>25</sup>Notice that the above limit for  $f''(x)$  simply comes from

$$f''(x) = \lim_{h \rightarrow 0} \frac{f'(x+h) - f'(x)}{h} = \lim_{h \rightarrow 0} \frac{\frac{f(x+h) - f(x)}{h} - \frac{f(x) - f(x-h)}{h}}{h} = \lim_{h \rightarrow 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}.$$

### 3.5. Example: Ornstein-Uhlenbeck process

---

By definition the infinitesimal generator is a linear operator; therefore  $\mathbf{A}$ , our FD approximation of  $\mathcal{A}$ , will be given by the sum of the FD approximation of the drift term and the FD approximation of the Ito correction term:

$$\frac{f_{i+1} - f_i}{\Delta x} \mu_{i,F}^+ + \frac{f_i - f_{i-1}}{\Delta x} \mu_{i,B}^- + \frac{1}{2} \frac{f_{i+1} - 2f_i + f_{i-1}}{(\Delta x)^2} \sigma_i^2$$

or, in matrix notation,

$$\underbrace{\begin{pmatrix} -\frac{\mu_{i,B}^-}{\Delta x} + \frac{1}{2} \frac{\sigma_i^2}{(\Delta x)^2} & \frac{\mu_{i,B}^-}{\Delta x} - \frac{\mu_{i,F}^+}{\Delta x} - \frac{\sigma_i^2}{(\Delta x)^2} & \frac{\mu_{i,F}^+}{\Delta x} + \frac{1}{2} \frac{\sigma_i^2}{(\Delta x)^2} \\ \text{inflow}_{i-1} \geq 0 & \text{outflow}_i \leq 0 & \text{inflow}_{i+1} \geq 0 \end{pmatrix}}_{i^{th} \text{ row of } \mathbf{A}} \begin{pmatrix} f_{i-1} \\ f_i \\ f_{i+1} \end{pmatrix}$$

where the terms in black are essentially the same as in chapter 2, while the terms in red are the new entries pertaining to our approximation of the correction term  $\frac{\sigma^2(x)}{2} f''(x)$ .

**Boundary Conditions** — In principle, we would be at this point ready to find a numerical solution to our KF eq. (3.27). However, just as an ODE might require initial conditions to have a unique solution, PDEs usually require boundary conditions (and possibly initial conditions) for uniqueness. In the absence of such conditions, a PDE might admit infinitely many solutions. We will discuss boundary conditions more in detail in the next chapter, suffice here to say that in the context of PDEs, boundary conditions express the behavior of the solution on the boundary of its domain.

Why then, were we able to ignore boundary conditions when finding the solution to the HJB equation in chapter 2? To answer this question it is worth noting that the problem we currently face is essentially that – because we are discretizing our state space onto a grid – we need to restrict the domain of our process, which in this example is  $(-\infty, \infty)$ , to a finite grid. When the process has no diffusion component ( $\sigma = 0$ ), as in chapter 2, explicit boundary conditions are sometimes unnecessary when the dynamics naturally keep the state within bounds – the deterministic drift pointed inward at both boundaries, and the upwind scheme automatically handled this by using forward differences at the lower boundary and backward differences at the upper boundary.

However, with diffusion ( $\sigma \neq 0$ ), the situation is fundamentally different. Even when the drift points inward at the boundaries (as in our OU process where the drift pulls toward the mean), the diffusion term creates probability flux at the boundaries that must be explicitly handled. Essentially, there is always a non-zero probability that a sufficiently large Brownian shock pushes the process outside our grid. Without proper boundary conditions, probability mass would “leak” out of our computational domain.

This creates a technical problem: the centered difference approximation of the diffusion term  $\frac{1}{2} \frac{\partial^2}{\partial x^2} [g(x)\sigma^2]$  requires values at  $i-1$  and  $i+1$ . At the boundaries  $i = 1$  and  $i = I$ , this would require “ghost nodes” at positions  $i = 0$  and  $i = I+1$  that lie outside our grid.<sup>26</sup> By

---

<sup>26</sup>Simply ignoring these diffusion terms at the boundaries would violate the requirement that rows of

imposing appropriate boundary conditions, we can specify how to handle probability mass that would otherwise escape the grid, allowing us to construct a well-defined generator matrix on our finite domain.

## Solving the KFE

We now have a way to write  $\mathbf{A}$  – our finite-difference approximation of  $\mathcal{A}$  – and, in order to find a FD approximation of its hermitian adjoint  $\mathcal{A}^*$  it is simply sufficient to take the transpose  $\mathbf{A}^T$ . To summarize, we have that  $\mathbf{A}$  is the discretized version of  $\mathcal{A}$ , and  $\mathbf{A}^T$  is the discretized version of  $\mathcal{A}^*$ .

Hence, the KF eq. (3.25) can be approximated as:

$$\frac{\mathbf{g}^{n+1} - \mathbf{g}^n}{\Delta} = \mathbf{A}^T \mathbf{g}^\bullet \quad (3.29)$$

where by  $\mathbf{g}^\bullet$  we simply mean that to solve the system we can use either an explicit method and replace  $\mathbf{g}^\bullet$  with  $\mathbf{g}^n$  on the right-hand side or an implicit method and replace it with  $\mathbf{g}^{n+1}$  – which have the same advantages and disadvantages as in chapter 2.

To find the stationary distribution such that  $0 = \mathcal{A}^* g$  we can then proceed in two ways: the more intuitive (though slower) way is to simply do time-iteration on the discretized eq. (3.29) starting from an initial guess  $\mathbf{g}^0$  until convergence – i.e. solve the initial-value problem – using either an implicit or explicit scheme.<sup>27</sup> This time-iteration approach works because, starting from any initial distribution, the system converges to its stationary distribution as  $t \rightarrow \infty$ .<sup>28</sup> In practice, one typically iterates until  $\|\mathbf{g}^{n+1} - \mathbf{g}^n\| < \varepsilon$  for some small tolerance  $\varepsilon$  (e.g.,  $10^{-6}$ ). The second approach, which is generally faster since it usually involves one single backslash operator, is to instead directly solve the eigenvalue problem  $\mathbf{A}^T \mathbf{g} = 0$ . While faster, this approach requires more care numerically. The key insight is that any stationary distribution must be in the null space of  $\mathbf{A}^T$ , which we can find by solving the eigenvalue problem. However, as we'll see, the singularity of  $\mathbf{A}$  requires special handling.

Given the FD approximation of  $\mathcal{A}^*$ , AT, the following code block shows how to solve the initial-value problem in eq. (3.29) using the implicit scheme:<sup>29</sup>

---

$\mathbf{A}$  sum to zero, which, as we saw in section 3.4.1, is a key property of infinitesimal generators.

<sup>27</sup>Note that since  $\mathbf{g}$  is a distribution, it must always sum to 1 (and so the starting distribution  $\mathbf{g}^0$  must also be a density). In theory it can be shown that the fact that the rows of  $\mathbf{A}$  sum to zero also implies mass preservation of  $\mathbf{g}$  throughout the time-iteration steps. In practice, however, the rows of  $\mathbf{A}$  might not sum *exactly* to zero due to numerical inaccuracies. It is therefore usually recommended to apply a normalization at each time step.

<sup>28</sup>Provided that: (i) a unique stationary distribution exists – which requires the continuous-time Markov process to be irreducible and positive recurrent – and (ii) the discretization preserves the key properties of the continuous operator, particularly that  $\mathbf{A}$  remains a valid generator matrix (rows sum to zero) and that, if using an explicit scheme, the time step  $\Delta$  is sufficiently small to satisfy the Courant-Friedrichs-Lowy condition for stability – specifically,  $\Delta \leq \frac{(\Delta x)^2}{\sigma_{\max}^2}$  where  $\sigma_{\max}^2$  is the maximum diffusion coefficient on the grid. For diffusion processes with reflecting boundaries, the first condition is typically satisfied when the drift points inward at the boundaries (ensuring the process doesn't get “stuck”) and the diffusion coefficient is positive in the interior of the domain (ensuring all states communicate); these are clearly satisfied for our OU process.

<sup>29</sup>This is essentially the same code that was used to generate fig. 3.5.

### 3.5. Example: Ornstein-Uhlenbeck process

---

```

1 # Initial value problem
2 for n = 1:N
3     D = SparseArrays.I - Δ*AT
4
5     # Solve system of equations
6     g[n + 1] = D\g[n]
7     g[n + 1] ./= sum(g[n + 1].*dx) # renormalize to ensure pdf sums to 1
8 end

```

---

If instead we are not actually interested in the evolution of  $g_t$  starting from some initial distribution  $g_0$  and we only want to find the stationary distribution, we can also directly solve the system  $\mathbf{A}^T \mathbf{g} = 0$ . This is equivalent to finding the eigenvector associated to the zero eigenvalue; however, exactly because the generator matrix  $\mathbf{A}$  has a zero eigenvalue, we also know that it is singular and we cannot just directly use our backslash operator on  $\mathbf{A}^T$ .<sup>30</sup> There are several ways to handle the singularity of  $\mathbf{A}$ , the approach we follow here essentially boils down to the following: first, “remove” an equation from our system so that we are left with  $I - 1$  equations in  $I$  unknowns; second, assign a value for one of the unknowns and solve the system for all others; finally, renormalize so that the density sums to 1.<sup>31,32</sup> “Essentially we remove one of the degrees of freedom by setting one of the elements of  $\pi$  equal to unity and later renormalize to get the correct stationary probability vector” (Stewart 2021, p. 75).

The next code block shows how to implement the “remove an equation” approach to solve for the stationary distribution in eq. (3.29):

```

1 # "Remove an equation" approach
2
3 # Choose any one equation to be removed, otherwise matrix is singular
4 i_fix = 10
5
6 # "Remove" equation from the system
7 AT[i_fix, :] .= 0
8 AT[i_fix, i_fix] = 1
9
10 # Fix "initial" solution for that unknown
11 b = zeros(Nx)
12 b[i_fix] = 1.0          # can be any value >0, will be renormalized
13
14 # Solve system of equations
15 g = AT\b
16 g ./= sum(g.*dx)        # renormalize to ensure pdf sums to 1

```

---

<sup>30</sup>Incidentally, a homogeneous system of  $n$  linear equations in  $n$  unknowns has a solution other than the trivial solution ( $x_i = 0 \forall i$ ) if and only if the coefficient matrix is singular. That is, the fact that  $\mathbf{A}$  is singular is specifically what ensures that  $\mathbf{A}^T \mathbf{g} = 0$  has a nontrivial solution.

<sup>31</sup>See Stewart (2021, section 2.3.1) for a detailed explanation of this approach as well as others

<sup>32</sup>The choice of which equation to remove (i.e., which  $i_{fix}$  to use) typically doesn’t matter for well-behaved problems with a unique stationary distribution. However, for numerical stability, it’s often best to choose an index where we expect the stationary distribution to have substantial mass.



# Chapter 4

## Stochastic HJB Equations

In this lecture we will extend the methodologies covered in chapter 2 to include uncertainty. At least in terms of techniques, this should be a relatively straightforward application of what we just learned in chapter 3. Specifically, after covering the general stochastic optimal control problem and stochastic HJB equations in section 4.1, we will directly apply these methods to a version of the Aiyagari model in continuous time. Just as in its discrete-time formulation, we will not be able to solve the model analytically and we will have to resort to numerical methods. We will then use the model to further explore the connection between HJB and KF equations. In particular, we will see that by solving the HJB equation we can directly get a solution to the KF equation, which obviously turns out to be particularly useful in heterogeneous-agent models in which the distribution is a key object of interest.<sup>1</sup>

### 4.1. Stochastic Optimal Control

Let's start by deriving the HJB equation for a generic stochastic optimal control problem in economics of the form:

$$v(x_0) = \max_{\alpha(t)} \mathbb{E}_0 \int_0^\infty e^{-\rho t} r(x(t), \alpha(t)) dt$$
$$\text{s.t. } dx(t) = f(x(t), \alpha(t))dt + \sigma(x(t))dW_t$$

for  $t \geq 0$  and  $x(0) = x_0$  given.

Variables are defined in the exact same way as in the deterministic problem we solved in lectures 1 and 2, while the terms in red highlight the differences between this problem and the deterministic one: first, the state variable  $x(t)$  now evolves stochastically due to the presence of the Brownian-motion term  $dW_t$  and, second, the value function will

---

<sup>1</sup>These notes have been written on the assumption that students are already familiar with the baseline Aiyagari model in discrete time (or, for all practical purposes, any standard incomplete market model). If this is not the case, there is plenty of readily available material covering that version of the model and its implications for consumption, savings, etc.

consequently be the maximum of the *expected value* of the utility function.<sup>2</sup> We will generally focus on the case in which the state variable  $x(t)$  is one-dimensional but, as we will see, the generalization to the multi-dimensional case is a simple application of multidimensional Itô's lemma.

In line with the heuristic approach we took in lecture 2, the **HJB equation** for this problem can immediately be written as:

$$\rho v(x) = \max_{\alpha} r(x, \alpha) + v'(x)f(x, \alpha) + \frac{1}{2}v''(x)\sigma^2(x) \quad (4.1)$$

where the first part is identical to what we had in the deterministic case, but now there is an additional term that depends on the second derivative of the value function and on the volatility of the state variable. At this point it should be easy to guess that this is nothing else than the usual variance correction term arising from applying Itô's lemma.

**Informal Derivation —** We derive the HJB equation by starting with the discrete-time formulation and taking the continuous-time limit. The discrete-time Bellman equation writes as:

$$v(x_t) = \max_{c_t} \Delta u(c_t) + e^{-\rho\Delta} \mathbb{E}_t [v(x_{t+\Delta})]$$

multiply by  $e^{\rho\Delta}$  and then subtract  $v(x_t)$  on both sides:

$$(e^{\rho\Delta} - 1) v(x_t) = \max_{c_t} e^{\rho\Delta} \Delta u(c_t) + \mathbb{E}_t [v(x_{t+\Delta}) - v(x_t)].$$

Dividing by  $\Delta$ , we obtain:

$$\frac{e^{\rho\Delta} - 1}{\Delta} v(x_t) = \max_{c_t} e^{\rho\Delta} u(c_t) + \frac{1}{\Delta} \mathbb{E}_t [v(x_{t+\Delta}) - v(x_t)]$$

which – recalling that, for small  $\Delta$ , we have  $\frac{e^{\rho\Delta} - 1}{\Delta} \approx \rho$  – we can rewrite in the limit of  $\Delta \rightarrow 0$  as:

$$\rho v(x_t) = \max_{c_t} u(c_t) + \frac{1}{dt} \mathbb{E}_t [dv(x_t)]. \quad (4.2)$$

At this point, we have essentially already derived the HJB, we are only left to figure out how to rewrite  $\mathbb{E}_t [dv(x_t)]$  in a more useful way, which we can do using Itô's lemma (since  $v$  is a function of  $x$  and the latter is a stochastic process):

$$\begin{aligned} dv &= \frac{\partial v}{\partial x} dx + \frac{1}{2} \frac{\partial^2 v}{\partial x^2} (dx)^2 \\ &= \left( v'(x)\mu(x) + \frac{1}{2}v''(x)\sigma^2(x) \right) dt + v'(x)\sigma(x)dW_t \end{aligned}$$

where we have used a general stochastic process  $dx = \mu(x)dt + \sigma(x)dW_t$ . Since  $\mathbb{E}_t[dW_t] = 0$ ,

---

<sup>2</sup>To simplify the problem (and in line with the models we will look at) we assume that the diffusion term  $\sigma(\cdot)$  does not depend on the control variable,  $\alpha(t)$ , but the problem trivially generalizes when we have  $\sigma(x_t, \alpha_t)$ .

#### 4.1. Stochastic Optimal Control

---

we have:

$$\mathbb{E}_t[dv(x_t)] = \left( v'(x)\mu(x) + \frac{1}{2}v''(x)\sigma^2(x) \right) dt$$

and, substituting in eq. (4.2), we finally get the HJB equation:

$$\rho v(x) = \max_c u(c) + v'(x)\mu(x) + \frac{1}{2}v''(x)\sigma^2(x). \quad (4.3)$$

Of course the expression in either eq. (4.1) or (4.3) will generally depend on the specific details of the problem. However, eq. (4.2) provides a rather flexible framework for deriving HJB equations in economics: for any problem with time-separable utility and exponential discounting, one can use (4.2) as a starting point and simply apply Itô's lemma to the specific state dynamics to obtain the complete HJB equation.

In fact, in case we have a vector of state variables  $\mathbf{x}$ , we can simply apply the multivariate version of Itô's lemma to find that, in such case, the HJB equation can be written as:

$$\rho v(\mathbf{x}) = \max_c u(c) + (\nabla_{\mathbf{x}}v)^T \boldsymbol{\mu} + \frac{1}{2}\text{tr}(\boldsymbol{\sigma}^T (\nabla_{\mathbf{x}}^2 v) \boldsymbol{\sigma})$$

where  $\boldsymbol{\mu}$  is a vector of drifts,  $\boldsymbol{\sigma}$  is a matrix of volatilities, and  $\nabla_{\mathbf{x}}v$  and  $\nabla_{\mathbf{x}}^2 v$  are the gradient and hessian of  $v$ , respectively.

**A Special Case —** Many economic problems often feature an exogenous state variable subject to shocks alongside a deterministic endogenous state variable. Given the prevalence of this structure, it is worth covering such specific case more in detail. Specifically, the state vector  $\mathbf{x} = (x_1, x_2)$  consists of an endogenous state variable  $x_1$  which typically lacks a stochastic component:<sup>3</sup>

$$dx_1 = f(x_1, x_2, \alpha)dt \quad (4.4)$$

while  $x_2$  is a stochastic exogenous state variable:

$$dx_2 = \mu(x_2)dt + \sigma(x_2)dW \quad (4.5)$$

Hence, we can immediately write:

$$\boldsymbol{\mu}(\mathbf{x}, \alpha) = \begin{pmatrix} f(x_1, x_2, \alpha) \\ \mu(x_2) \end{pmatrix}, \quad \boldsymbol{\sigma}(\mathbf{x}) = \begin{pmatrix} 0 \\ \sigma(x_2) \end{pmatrix}$$

and the HJB equation is given by:

$$\rho v(x_1, x_2) = \max_{\alpha} r(x_1, x_2, \alpha) + v_1(x_1, x_2)f(x_1, x_2, \alpha) + v_2(x_1, x_2)\mu(x_2) + \frac{1}{2}v_{22}(x_1, x_2)\sigma^2(x_2)$$

subject to eqs. (4.4) and (4.5).

---

<sup>3</sup>To simplify notation we are omitting the time subscript.

### 4.1.1. Example: The RBC Model

As a concrete example, let's extend the neoclassical growth model from Chapter 2 by making total factor productivity stochastic. This transforms our deterministic framework into the canonical real business cycle model, allowing us to study how productivity shocks propagate through capital accumulation.

The sequential formulation of the model is:

$$v(k_0, z_0) = \max_{\{c_t\}_{t \geq 0}} \mathbb{E}_0 \int_0^\infty e^{-\rho t} u(c(t)) dt$$

$$\text{s.t. } \begin{cases} dk = (zF(k) - \delta k - c)dt \\ dz = \mu(z)dt + \sigma(z)dW \end{cases}$$

with  $k(0) = k_0$  and  $z(0) = z_0$  given. Here capital  $k$  is the endogenous state variable, TFP  $z$  is the exogenous stochastic state, and consumption  $c$  is the control variable.

Following our general framework, the HJB equation for this problem is:

$$\rho v(k, z) = \max_c u(c) + v_k(k, z)(zF(k) - \delta k - c) + v_z(k, z)\mu(z) + \frac{1}{2}v_{zz}(k, z)\sigma^2(z) \quad (4.6)$$

Comparing equation (4.6) with the deterministic case from Chapter 2, we see that uncertainty affects the value function through the terms  $v_z\mu(z) + \frac{1}{2}v_{zz}\sigma^2(z)$ . The first captures the expected growth of TFP, while the second represents the value of holding capital as a hedge against productivity risk. This hedging motive will generally increase the steady-state capital stock relative to the deterministic case.

The choice of stochastic process for TFP has important economic and computational implications. We consider three common specifications:

**Geometric Brownian Motion**  $dz = \mu z dt + \sigma z dW$  — This ensures TFP remains positive and has the convenient property that  $\log z$  follows a normal distribution, a common modeling assumption.

**Mean-Reverting Process**  $dz = \eta(\bar{z} - z)dt + \sigma dW$  — Possibly the most common specification of the RBC model is to assume that the log of TFP follows an Ornstein-Uhlenbeck process, which ensures TFP remains positive for appropriate parameter values. In that case, it is often better (especially computationally) to use  $\log z$  as the actual state variable. Alternatively, one can also model TFP to follow a Cox-Ingersoll-Ross (CIR) process

$$dz = \eta(\bar{z} - z)dt + \sigma\sqrt{z}dW$$

which also ensures that TFP remains positive.

**Continuous-Time Markov Chain** — This provides a tractable way to model regime switches (e.g., recessions vs. expansions) and will be particularly useful in our study of

## 4.2. Boundary Conditions

---

heterogeneous agent models. Note that if  $z$  follows a two-states CTMC  $z \in \{z_1, z_2\}$ , the HJB equation takes the form:

$$\rho v_i(k) = \max_c u(c) + v'_i(k) (z_i F(k) - \delta k - c) + \lambda_i [v_j(k) - v_i(k)] \quad \text{for } i = 1, 2, j \neq i.$$

The intuition is straightforward: when in state  $i$ , the household transitions to state  $j$  at rate  $\lambda_i$ , gaining  $v_j(k) - v_i(k)$  in value terms.

The table below summarizes how each specification enters the HJB equation:

Process	Drift $\mu(z)$	Diffusion $\sigma^2(z)$	Advantage
Geometric BM	$\mu z$	$\sigma^2 z^2$	Log-normality
OU or CIR	$\eta(\bar{z} - z)$	$\sigma^2$ or $\sigma^2 z$	Mean reversion
CTMC	$\lambda_z[z' - z]$	0	Discrete regimes

While the RBC is a relatively straightforward extension of the neoclassical growth model studied in the first two chapters and as such a very useful pedagogical example, we will momentarily turn to a more interesting application (at least for the application of continuous-time methods): the Aiyagari model. Before doing so, however, it is useful to discuss boundary conditions more in depth.

## 4.2. Boundary Conditions

Before moving on to solve the Aiyagari model, it is important to first discuss boundary conditions. This is because in order to pin down a unique solution to the HJB and KF equations – like most PDEs – one generally needs to add sufficient auxiliary information, and boundary conditions are one possible source of such information (and in fact the most common in economics). In fact, it is usually the case that a solution to these equations can only be defined together with appropriate boundary conditions, and that different boundary conditions may lead to different solutions. Hence, being able to understand what boundary conditions are and how to apply them is fundamental for solving these models.

Broadly speaking, boundary conditions are simply conditions that need to be added to our HJB and KF equations and that determine how the solution to such equations must behave at the boundary of the state space. As such, just as in an initial-value problem we can use initial (or terminal) conditions to pin down the unique solution to ODEs, in a boundary-value problem we will use boundary conditions to pin down the unique solution to a PDE.

There are several types of boundary conditions, many of which are seldomly used in economics; here we will therefore only look at some of the most common ones. Specifically, the type of conditions that are most often encountered in heterogeneous-agent models are so-called **state constraints**, which essentially boil down to imposing a restriction on the policy functions of our problem guaranteeing that the process does not abandon the domain. Other “classic” types of boundary conditions are **absorbing** and **reflecting** bound-

aries, also called Dirichlet and Neumann boundary conditions, respectively. Informally, we say that a stochastic process  $X$  has an absorbing boundary if  $X$  is removed from the system once the boundary is reached, and that it has a reflecting boundary if, when  $X$  hits the boundary, it is reflected back into the interior of the space.

In economic applications, reflecting boundaries typically represent constraints that agents cannot cross – for example, a borrowing constraint or a capacity constraint in production. Absorbing boundaries, on the other hand, represent exit points where agents leave the system permanently upon reaching them – such as bankruptcy or firm-exit thresholds. In heterogeneous agent models – in particular when working with distributions – reflecting boundaries are particularly common since we often want to study environments where the total mass of agents is constant over time (whereas in the absence of entry conditions absorbing boundaries drain mass from the system).

At this point it should therefore be clear that, while boundary conditions are needed in order to have a well-posed problem, they also have economic meaning: they encode what happens when agents reach the limits of their feasible choices.

**HJB vs KFE** — Even though the HJB and KF equations are tightly connected through the adjoint operator relationship discussed in chapter 3, they require complementary boundary conditions that reflect their different roles. The HJB is a backward equation that determines the value function and, as such, boundary conditions for the HJB specify either the *value* or the *marginal value* when certain constraints bind. On the other hand, the KFE is a forward equation that determines the evolution of a distribution and, as such, boundary conditions for the KFE specify how the *probability mass* behaves at the boundary.

Notice, however, that this difference in the treatment of boundary conditions is not arbitrary: for the adjoint relationship between the HJB operator  $\mathcal{A}$  and the KFE operator  $\mathcal{A}^*$  to hold, the boundary integrals that arise from integration by parts must vanish, which dictates how boundary conditions must be paired. For example, a reflecting boundary (where the state cannot exit) requires a zero-flux condition in the KFE (no probability mass escapes) paired with a Neumann-type condition in the HJB (specifying the derivative of the value function).

While subtle, this distinction is key: HJB boundary conditions affect individual optimization, KFE boundary conditions ensure mass preservation, and the two must be consistent through the adjoint relationship.

### 4.2.1. Probability Flux and Conservation

To better understand the role of boundary conditions, particularly in the context of KF equations, one useful concept is that of probability flux. Consider a general diffusion process:

$$dX = \mu(X)dt + \sigma(X)dW \tag{4.7}$$

## 4.2. Boundary Conditions

---

with associated KFE:

$$\partial_t g(x, t) = -\partial_x[\mu(x)g(x, t)] + \frac{1}{2}\partial_{xx}[\sigma^2(x)g(x, t)]. \quad (4.8)$$

The **probability flux**  $J(x, t)$  measures how much probability mass is moving past point  $x$  per unit time and can be written as:

$$J(x, t) = \mu(x)g(x, t) - \frac{1}{2}\partial_x[\sigma^2(x)g(x, t)]. \quad (4.9)$$

To understand this, imagine tracking the fraction of agents crossing wealth level  $x$ : if  $J(x, t) > 0$  the net movement is rightward (toward higher  $x$ ); if  $J(x, t) < 0$  the net movement is leftward (toward lower  $x$ ); finally, if  $J(x, t) = 0$  there is no net movement (i.e., there are equal flows in both directions).

The probability flux naturally has two components: a drift component,  $\mu(x)g(x, t)$ , which tells us that if agents at position  $x$  have drift  $\mu(x)$ , then the mass of agents crossing  $x$  due to deterministic movement is  $\mu(x)g(x, t)$  (think of this as “directed movement”: if everyone at wealth level  $x$  saves at rate  $\mu(x)$ , this term captures how many are moving past  $x$ ); and a diffusion component,  $-\frac{1}{2}\partial_x[\sigma^2(x)g(x, t)]$ , which tells us that random shocks cause agents to move both left and right (on net, diffusion moves probability from regions of high concentration to low concentration, like heat spreading from hot to cold).

The KFE can now be written as the conservation law:

$$\partial_t g(x, t) + \partial_x J(x, t) = 0, \quad (4.10)$$

which tells us that the change in probability at any point equals the net flow into that point. Since probability cannot be created or destroyed, any decrease at one location must be balanced by increases elsewhere. At the boundaries, we will need to specify what happens to the flux (agents who would otherwise exit the domain) – do they bounce back (reflecting), disappear (absorbing), or are simply prevented from reaching the boundary by the optimal policy?

### 4.2.2. Types of Boundary Conditions

As mentioned earlier, the most common types of boundary conditions encountered in economics are reflecting boundaries, absorbing boundaries, and state constraints. As their name suggest, reflecting boundary conditions are such that if a particle hits the boundary of the domain it is “reflected” back to its interior, while absorbing boundary conditions are such that if the process reaches the boundary of the domain it will be “absorbed” by it. State constraints on the other hand are not boundary conditions in the strict sense, but rather restrictions that the state variables must satisfy. Nonetheless, these restrictions essentially transform what would be a complementary slackness problem in discrete time into a boundary value problem in continuous time and, as such, they are closely related to boundary conditions. The key insight for continuous time methods is that constraints which “occasionally bind” in discrete time manifest exclusively as boundary conditions in

continuous time, greatly simplifying both the theoretical and computational treatment.<sup>4</sup>

**State Constraints** — State constraints require that the state variable remains within a feasible domain – for example, a borrowing constraint  $a \geq \underline{a}$  in a standard incomplete-markets model. In continuous time, such constraints never bind in the interior of the state space; they only matter at the boundary itself.<sup>5</sup>

Mathematically, when the state is constrained to remain in its domain, the constraint manifests as a **boundary inequality** on the value function. For our borrowing constraint example, at the boundary the value function must satisfy  $v'(\underline{a}) \geq u'(y + r\underline{a})$  which ensures that the optimal savings policy satisfies  $s(\underline{a}) \geq 0$  (i.e. that the drift points inward at the boundary).

The economic intuition is straightforward: at the constraint boundary, agents face asymmetric choices. They can move into the interior (save) but cannot move further out (borrow beyond the limit). The boundary inequality ensures that the marginal value of wealth is high enough that agents choose not to violate the constraint. When the inequality holds with equality, agents are indifferent between staying at the boundary or moving slightly into the interior; when it holds strictly, agents actively choose to move away from the constraint.

Importantly, this boundary inequality is a *Neumann-type* condition (i.e. involving derivatives) rather than a Dirichlet condition (i.e. specifying values). This is because the value function itself is not fixed at the boundary – only its behavior is constrained to ensure feasibility. In the KFE, the corresponding condition would be that probability flux through the boundary must be zero, ensuring mass conservation. Importantly, however, when the optimal behavior from the HJB equation naturally keeps the state away from the boundary (i.e. when  $s(\underline{a}) \geq 0$ ), the KFE requires no explicit boundary condition at that point – the zero-flux condition emerges automatically from the optimal policy.<sup>6</sup>

**Reflecting Boundaries (Neumann Conditions)** — Reflecting boundaries arise (or are imposed) when agents cannot cross boundaries but the natural evolution of the underlying process might push them there.

As an example, for the HJB equation, a reflecting boundary at  $x = \underline{x}$  would impose a Neumann boundary condition like:

$$v'(\underline{x}) = \bar{v} \tag{4.11}$$

---

<sup>4</sup>Slightly more formally, state constraints in optimal control problems manifest through boundary inequalities rather than standard boundary conditions. The interested reader can check exactly how to handle state constraints and how these connect to boundary conditions in the supplementary material to Achdou et al. (2022) and references therein.

<sup>5</sup>The intuition for why such constraints only show up as boundary conditions is simple: because time is continuous, if the state variable is such that the constraint is not binding at time  $t$  (e.g. wealth is strictly above the borrowing constraint), it will remain not binding an infinitesimal time interval later.

<sup>6</sup>In practical terms for economists implementing finite difference methods, the Neumann-type boundary condition appears naturally through the upwind scheme. At a borrowing constraint, the backward difference at the boundary uses the boundary condition  $v'(\underline{a}) = u'(y + r)$  only when the drift would otherwise be negative. The forward difference is computed normally. The scheme automatically selects which difference to use based on the sign of optimal savings, ensuring the boundary condition is used precisely when needed to prevent constraint violations.

## 4.2. Boundary Conditions

---

where  $\bar{v}$  is some pre-specified value. In economic applications, the specified value typically comes from optimality conditions – for instance, from the FOC at a borrowing constraint:  $v'(\underline{a}) = u'(r\underline{a} + y)$ . On the other hand, for the KFE, the corresponding condition would impose a zero flux condition like:

$$J(\underline{x}) = 0, \quad J(\bar{x}) = 0 \quad (4.12)$$

which ensures that no probability mass exits the domain.

The zero flux condition in eq. (4.12) often translates to setting  $g'(\underline{x}) = g'(\bar{x}) = 0$  when the drift is zero at the boundary. To see where this is coming from, recall the definition of the flux:

$$J(x) = \mu(x)g(x) - \frac{1}{2}\partial_x[\sigma^2(x)g(x)].$$

When the drift  $\mu(x)$  is zero at the boundary (and volatility is constant), this simplifies to  $g'(x) = 0$ , meaning the density is flat at the boundary. To gain some intuition about why this implements a reflecting boundary, consider what  $g'(\cdot)$  represents. If  $g(x)$  is the probability density at point  $x$ , setting  $g'(\underline{x}) = 0$  means that the density is flat at the boundary – it neither increases nor decreases. If probability mass were to accumulate at the boundary (as would happen if the process got “stuck” there), we would see  $g$  increasing sharply near the boundary. If instead mass were escaping the domain, we would see  $g$  decreasing toward zero at the boundary. By forcing  $g'(\underline{x}) = 0$  we ensure that any probability mass reaching the boundary is neither accumulated nor lost, but rather “bounced back” into the interior of the domain – hence the term “reflecting.”<sup>7,8</sup>

**Absorbing Boundaries (Dirichlet Conditions)** — While reflecting boundaries are more common in economic applications, absorbing boundaries might still arise in models with death states, market exit, or terminal conditions.<sup>9</sup> For absorbing boundaries, the HJB equation receives a Dirichlet condition (i.e. specified value:  $v(x)|_{\partial\mathcal{X}} = \bar{v}$ ), while the KFE equation has zero density at the boundary ( $g(x)|_{\partial\mathcal{X}} = 0$ ) where  $\bar{v}$  could represent, for instance, the value of exiting (liquidation value, scrap value, or bankruptcy cost).

### 4.2.3. Practical Implementation

In standard incomplete market models in continuous time, the lower bound (borrowing constraint at  $a = \underline{a}$ ) is usually modeled as a state constraint, which we showed the upwind scheme naturally takes care of; while the upper bound  $a = \bar{a}$  is often just artificially set to a value that is sufficiently large to ensure that  $s(\bar{a}) < 0$  which is therefore again

---

<sup>7</sup>More formally, this boundary condition ensures that the probability flux through the boundary is zero, meaning no probability mass crosses the boundary.

<sup>8</sup>The pairing of Neumann conditions on the HJB equation with zero-flux conditions on the KFE is not arbitrary – it ensures that the adjoint relationship between the two operators is preserved. When performing integration by parts to verify this relationship, these paired conditions guarantee that boundary terms vanish.

<sup>9</sup>In fact, truly absorbing boundaries are actually very rare – most “exit” states usually either allow re-entry or have continuation values.

taken care of by the upwind scheme. Nonetheless, in the presence of stochastic elements one needs to be extra careful since, for example, the presence of a diffusion term might make boundary conditions necessary. Intuitively, while the drift component of the process naturally respects the upwind scheme, the diffusion term introduces random variation that can push probability mass toward boundaries even when the drift points away from them. Without explicit boundary conditions – typically implemented as reflecting barriers – the second-order differential operator would be ill-defined at the boundaries, and probability mass could artificially “leak” out of the state space. Economically, reflecting barriers ensure that agents who hit either the borrowing constraint  $\underline{a}$  or the upper bound  $\bar{a}$  (due to idiosyncratic shocks) remain in the feasible state space, with their wealth dynamics appropriately constrained at these boundaries.

### Reflecting boundaries

Given the prevalence of reflecting boundary conditions in economics, especially when it comes to solving KF equations, it is worthwhile spending some time addressing how to implement them in our algorithms.

As we saw, imposing a reflecting boundary condition boils down to specifying a value for the derivative of  $f$  at the boundary. Specifically, let us assume that the derivative of the function at the boundaries is constant (zero in our case):  $f'(\underline{x}) = f'(\bar{x}) = 0$ .

In terms of numerical solution this implies that:

- ▶ Using the forward difference approximation of the derivative at the *beginning* of the state-space:

$$f'(x_1) = 0 \approx \frac{f_2 - f_1}{\Delta x} = 0 \implies f_1 = f_2$$

- ▶ Using the backward difference approximation of the derivative at the *end* of the state-space:

$$f'(x_I) = 0 \approx \frac{f_I - f_{I-1}}{\Delta x} = 0 \implies f_{I-1} = f_I$$

which can be imposed by ensuring that the first two entries of  $\mathbf{A}$  (at indices  $(1, 1)$  and  $(1, 2)$ ) as well as its last two entries (at indices  $(I, I-1)$  and  $(I, I)$ ) are equal to each other. In fact, by doing so, for any  $\beta \neq 0$  the first row of our approximation to eq. (3.26) would read

$$\begin{pmatrix} -\beta & \beta & 0 & \dots \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \end{pmatrix} = -\beta(f_1 - f_2)$$

and equivalently for the last row. When transposed and set to zero in the KFE, this then ensures that  $f_1$  is equal to  $f_2$  (and  $f_{I-1} = f_I$ ) and that our reflecting boundary conditions are satisfied in the numerical approximation.

## 4.3. Aiyagari in Continuous-Time

### 4.3.1. Setup

The setup of the model in continuous time is almost identical to the one in discrete time (Aiyagari 1994). There is a continuum of mass 1 of infinitely-lived households with CRRA utility and exponential discounting. Households inelastically supply 1 unit of time to the labour market and are endowed with  $y_t$  efficiency units of labour which follow a two-state CTMC with intensities  $\lambda_1$  and  $\lambda_2$ .<sup>10</sup> There is a representative firm with Cobb-Douglas technology  $Y_t = F(K_t, L_t) = K_t^\alpha L_t^{1-\alpha}$ . Markets for goods and factors of production are perfectly competitive: capital and labor are traded at prices  $r_t$  and  $w_t$ , respectively, while the price of output is normalized to 1. Finally, capital depreciates at rate  $\delta$ . The firm's problem is therefore the classic profit maximization problem of the form:<sup>11</sup>

$$\max_{K,L} F(K, L) - wL - (r + \delta)K.$$

Going back to the household side, the budget constraint is given by the exact counterpart of the discrete-time budget constraint  $a_{t+1} = (1 + r_t)a_t + y_t - c_t$ , that is:

$$\dot{a} = wy + ra - c$$

where savings,  $\dot{a}$ , obviously depend on labor income  $wy$ , capital income  $ra$ , and consumption  $c$ . Households face an exogenous borrowing constraint:

$$a \geq -\phi$$

where  $\phi \geq 0$  is the minimum level of assets that a household can have at all times.

### 4.3.2. HJB and KF Equations

Using the same approach as in section 4.1, we can immediately write the household's HJB equation as:

$$\rho v(a, y) = \max_c u(c) + v_a(a, y) (wy + ra - c) + \lambda_y [v(a, y') - v(a, y)], \quad (4.13)$$

---

<sup>10</sup>In the original article, Aiyagari (1994) assumes that  $y_t$  follows an AR(1) process which is then discretized using a 5-states discrete time Markov chain. In this sense a two-state CTMC is a natural simplification of the original model. Modifying the model to either feature an Ornstein-Uhlenbeck process or add more states to the CTMC is straightforward.

<sup>11</sup>The Aiyagari model is only one of the possible specifications of the so-called standard incomplete-markets (SIM) model (see Heathcote et al. 2009). Other very common specifications are the Bewley model (Bewley 1977) and the Huggett model (Huggett 1993). While the household blocks of these models are virtually identical, the main difference lies in the treatment of the supply side of the economy: the Aiyagari model has capital in positive net supply (and a production block); the Bewley model has risk-free government bonds in positive net supply (and a government block); and the Huggett model has assets in zero net supply (agents trade IOUs). While they all have their differences, these can be safely ignored when it comes to solving the household problem, which is what we are mainly concerned about.

while the KFE is instead given by:

$$\partial_t g_t(a, y_j) = \underbrace{-\partial_a [s(a, y_j) g_t(a, y_j)]}_{\text{Wealth changes}} + \underbrace{\lambda_{j'} g_t(a, y_{j'}) - \lambda_j g_t(a, y_j)}_{\text{Income changes}} \quad \text{for } j = 1, 2, j' \neq j \quad (4.14)$$

where  $g_t(a, y)$  is the cross-sectional distribution over wealth and income.

Since in chapter 3 we mostly focused on the KFE for univariate diffusion processes while here we have a multivariate KFE in which one of the state variables is a CTMC, it is useful to look at eq. (4.14) more in detail: as usual, the change in the distribution  $g_t$  over time is given by the “transition rates” of its states. Since there are now two states – wealth  $a$  and income  $y$  – the KFE accounts for how each changes. For wealth changes, we have essentially the same term as in the KFE for a generic diffusion, where here we are simply denoting the drift of  $a$  as  $s(a, y_j)$  for savings. For income changes, we have the same terms we had when looking at the KF eq. (3.20) for CTMC: households in state  $y_j$  transition to state  $y_{j'}$  at rate  $\lambda_j$  so a mass  $\lambda_j g_t(a, y_j)$  leaves state  $y_j$ , and households in state  $y_{j'}$  transition to state  $y_j$  at rate  $\lambda_{j'}$  so a mass  $\lambda_{j'} g_t(a, y_{j'})$  arrives in state  $y_j$ .

With just two states,  $y_1$  and  $y_2$ , we can also rewrite the same KFE as:

$$\partial_t g_t(a, y_1) = -\partial_a [s(a, y_1) g_t(a, y_1)] \quad \underbrace{-\lambda_1 g_t(a, y_1)}_{\text{outflow: } y_1 \rightarrow y_2} \quad \underbrace{+\lambda_2 g_t(a, y_2)}_{\text{inflow: } y_2 \rightarrow y_1} \quad (4.15)$$

$$\partial_t g_t(a, y_2) = -\partial_a [s(a, y_2) g_t(a, y_2)] \quad \underbrace{-\lambda_2 g_t(a, y_2)}_{\text{outflow: } y_2 \rightarrow y_1} \quad \underbrace{+\lambda_1 g_t(a, y_1)}_{\text{inflow: } y_1 \rightarrow y_2} \quad (4.16)$$

which is exactly identical but possibly more intuitive: households in state 1 move to state 2 at rate  $\lambda_1$ , and therefore  $\lambda_1 g(a, y_1)$  has negative sign in eq. (4.15) and positive sign in eq. (4.16); viceversa, households in state 2 move to state 1 at rate  $\lambda_2$ , and therefore  $\lambda_2 g(a, y_2)$  has negative sign in eq. (4.16) and positive sign in eq. (4.15).

Finally, to solve for the stationary distribution, we simply impose that the derivatives with respect to time on the left-hand side of eqs. (4.15) and (4.16) are both equal to 0 to get:

$$\begin{aligned} 0 &= -\partial_a [s_1(a) g_1(a)] - \lambda_1 g_1(a) + \lambda_2 g_2(a) \\ 0 &= -\partial_a [s_2(a) g_2(a)] - \lambda_2 g_2(a) + \lambda_1 g_1(a) \end{aligned}$$

where we simplified notation by writing  $x_j(a) = x(a, y_j)$ . At the cost of being repetitive, the first term is the same as in the case of Brownian motion where  $s_j(a)$  is the drift of  $a$ , while the second term represents the transition across income states.

Having already discussed adjoint operators, it is useful to highlight the tight connection between the HJB and KF equations. In fact, we can rewrite the two more compactly as:

$$\text{HJB: } \rho v = \max_c u(c) + \mathcal{A}v$$

$$\text{KFE: } \partial_t g = \mathcal{A}^* g$$

where the KFE operator  $\mathcal{A}^*$  is the Hermitian adjoint of the HJB operator  $\mathcal{A}$  given by: More formally, these two equations constitute a forward-backward system of coupled PDEs.<sup>12</sup>

In practice, in order to solve these two equations, we will be using finite difference methods. Since these methods boil down to approximating the infinitesimal generators using matrices and since, as discussed in the previous lecture, in finite dimensions the Hermitian adjoint is equivalent to the matrix transpose, we will therefore be able to write the system in matrix notation as:

$$\begin{aligned}\text{HJB: } \rho\mathbf{v} &= \mathbf{u} + \mathbf{Av} \\ \text{KFE: } \partial_t\mathbf{g} &= \mathbf{A}^T\mathbf{g}\end{aligned}$$

where  $\mathbf{A}^T$  is the transpose of matrix  $\mathbf{A}$ .

### 4.3.3. Equilibrium

To close the model, we are only left to define the equilibrium. A stationary Recursive Competitive Equilibrium (RCE) of the Aiyagari economy is a list of:

1. Prices  $r$  and  $w$
2. Value function  $v(a, y)$  and associated policy functions  $c(a, y), s(a, y)$
3. Aggregate capital stock  $K$  and labor supply  $L$
4. Stationary distribution  $g(a, y)$

such that:

1. **Agents' optimization:** Given  $(r, w)$ , the functions  $v(\cdot), c(\cdot), s(\cdot)$  satisfy the HJB:

$$\rho v(a, y_j) = \max_c u(c) + v_a(a, y_j) (wy_j + ra - c) + \lambda_{j,j'} [v(a, y_{j'}) - v(a, y_j)]$$

2. **Firm's optimization:** Given  $(r, w)$ , factor demands  $K$  and  $L$  solve the firm FOCs

3. **Consistency:** The stationary distribution  $g(a, y)$  satisfies the KFE:

$$0 = -\partial_a [s(a, y_j)g(a, y_j)] + \sum_{j'} \lambda_{j',j} g(a, y_{j'}) - g(a, y_j) \sum_{j'} \lambda_{j,j'} \quad \forall j$$

4. **Market clearing:** The markets for capital and labor clear:

$$K = \sum_j \int_a ag(a, y_j)da \quad L = \sum_j \int_a y_j g(a, y_j)da$$

---

<sup>12</sup>Forward-backward because the HJB is a backward equation which is solved backward in time starting from a terminal condition, while the KFE is a forward equation which is solved forward in time using an initial condition.

Where the consistency condition – which here requires that the distribution is stationary – is nothing else than the continuous-time counterpart of the usual condition you get in discrete-time models asking that the operator maps the distribution into itself.

## 4.4. Numerical Solution

### 4.4.1. Upwind Scheme

As usual, we will solve the model by approximating the HJB equation using finite differences. Recall the HJB equation:

$$\rho v_j(a) = \max_c u(c) + v'_j(a) (wy_j + ra - c) + \lambda_j [v_{j'}(a) - v_j(a)]$$

and what we do is to approximate the value function  $v_j(a) = v(a, y_j)$  at discrete points  $a_i \in \{a_i\}_{i=1}^I$ . We denote  $v(a_i, y_j)$  as  $v_{i,j}$  and the forward and backward approximations of the derivative of the value function  $v'_j(a) = v'(a, y_j)$  as:

$$\begin{aligned} v'_{i,j,B} &= \frac{v_{i,j} - v_{i-1,j}}{\Delta a} \\ v'_{i,j,F} &= \frac{v_{i+1,j} - v_{i,j}}{\Delta a} \end{aligned}$$

where  $\Delta a$  is the step size between the points on the asset grid. Given these premises, the discretized HJB is given by:

$$\rho v_{i,j} = u(c_{i,j}) + v'_{i,j} \cdot (wy_j + ra_i - c_{i,j}) + \lambda_j (v_{i,j'} - v_{i,j})$$

where  $c_{i,j}$  is the consumption at the point  $a_i$  and  $y_j$ .

Just as we did in lecture 2, in order to choose which approximation to use for  $v'_{i,j}$  we will be using an upwind scheme on the drift of  $a$ . Depending on the choice of approximation, the drift of  $a$  is then given by:

$$\begin{aligned} \text{Forward Drift: } s_{i,j,F} &= wy_j + ra_i - (u')^{-1}(v'_{i,j,F}) \\ \text{Backward Drift: } s_{i,j,B} &= wy_j + ra_i - (u')^{-1}(v'_{i,j,B}) \end{aligned}$$

where as usual the difference between the two is in the use of the forward or backward approximation of  $v'$  in the FOC. The “upwind” derivative is then defined as:

$$v'_{i,j} = v'_{i,j,F} \cdot \mathbf{1}_{\{s_{i,j,F} > 0\}} + v'_{i,j,B} \cdot \mathbf{1}_{\{s_{i,j,B} < 0\}} + \bar{v}'_{i,j} \cdot \mathbf{1}_{\{s_{i,j,F} < 0 < s_{i,j,B}\}}$$

where  $\bar{v}'_{i,j} = u'(wy_j + ra_i)$  (i.e.  $\dot{a} = 0$ ). That is, as it should hopefully be somewhat familiar by now, the upwind scheme uses the forward difference when the forward drift is positive, the backward difference when the backward drift is negative, and the “steady state” when the forward drift is negative and the backward drift is positive (which, as long as the value function is concave, covers all possible cases).

#### 4.4. Numerical Solution

---

We can therefore write the upwind scheme approximation of the HJB as:

$$\rho v_{i,j} = u(c_{i,j}) + v'_{i,j,F} \cdot s_{i,j,F}^+ + v'_{i,j,B} \cdot s_{i,j,B}^- + \lambda_j (v_{i,j'} - v_{i,j})$$

or, in matrix notation, as:

$$\rho \mathbf{v} = \mathbf{u} + \mathbf{A}\mathbf{v}$$

where  $\mathbf{v}$  is now a vector of length  $I \cdot J$  and  $\mathbf{A}$  is an  $(I \cdot J \times I \cdot J)$  matrix, which is the finite-difference representation of the infinitesimal generator of  $v$ ,  $\mathcal{A}$ , and which contains “transitions” across all wealth and income pairs.

#### 4.4.2. Matrix Representation

Because the generator matrix  $\mathbf{A}$  needs to account for transitions across wealth-income pairs, it also needs to contain the generator of the associated earnings process which – in this simple case with only two income states – is given by:

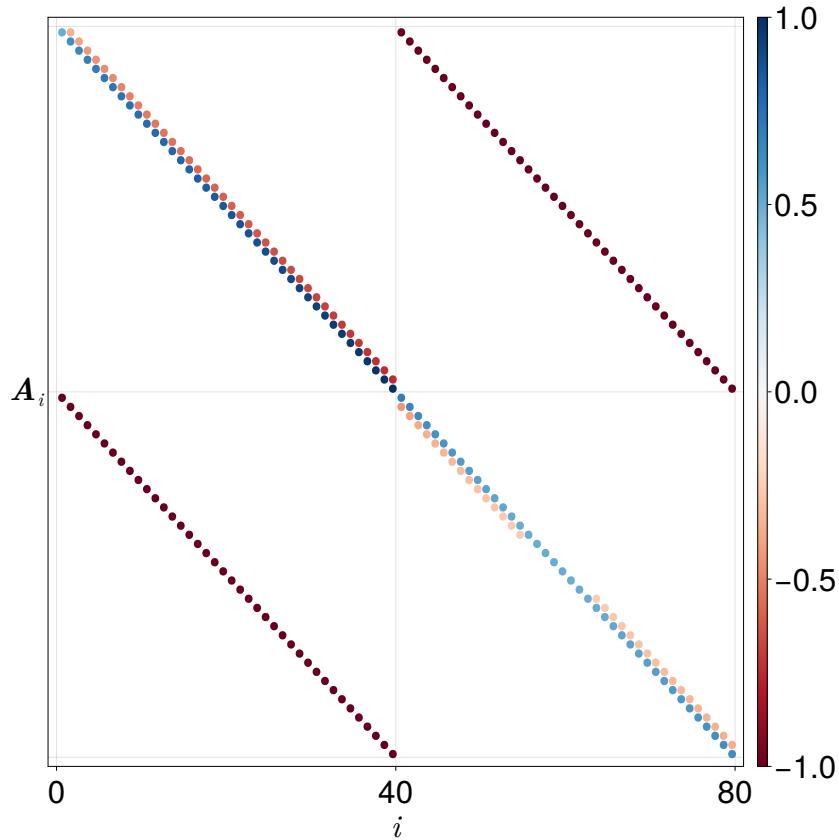
$$\Lambda = \begin{pmatrix} -\lambda_1 & \lambda_1 \\ \lambda_2 & -\lambda_2 \end{pmatrix}.$$

Recall in fact that households in the first income state transition to the second income state at rate  $\lambda_1$  and households in the second income state transition to the first at rate  $\lambda_2$ .

Because  $\mathbf{A}$  is now going to be of size  $(I \cdot J)^2$ , if we order first by wealth and then income (i.e. elements of  $\mathbf{A}$  are indexed  $i,j$ ), every row associated with  $j = 1$  is going to have  $-\lambda_1$  on the main diagonal and  $+\lambda_1$  on the column associated with  $j = 2$ , while every row associated with  $j = 2$  will have  $-\lambda_2$  on the main diagonal and  $+\lambda_2$  on the column associated with  $j = 1$ . That is, ignoring for a second the terms associated with the drift of  $a$  (which are essentially the same as in lecture 2), the elements of  $\mathbf{A}$  associated with earnings switching are:

$$\mathbf{A}_y = \left( \begin{array}{cc|ccccc} -\lambda_1 & 0 & 0 & \cdots & 0 & \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & -\lambda_1 & 0 & \cdots & 0 & 0 & \lambda_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -\lambda_1 & 0 & 0 & \cdots & 0 & \lambda_1 & 0 \\ 0 & \cdots & \cdots & 0 & -\lambda_1 & 0 & \cdots & \cdots & 0 & \lambda_1 \\ \hline \lambda_2 & 0 & 0 & \cdots & 0 & -\lambda_2 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 & 0 & -\lambda_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_2 & 0 & 0 & \cdots & 0 & -\lambda_2 & 0 \\ 0 & \cdots & \cdots & 0 & \lambda_2 & 0 & \cdots & \cdots & 0 & -\lambda_2 \end{array} \right)$$

and our big matrix can be written as  $\mathbf{A} = \mathbf{A}_a + \mathbf{A}_y$  where  $\mathbf{A}_a$  is essentially the same as



**Figure 4.1:** Structure of  $\mathbf{A}$

in lecture 2:

$$\underbrace{\left( \begin{array}{cccccc} 0 & \dots & -\frac{s_{i,j,B}^-}{\Delta a} & \frac{s_{i,j,B}^-}{\Delta a} - \frac{s_{i,j,F}^+}{\Delta a} & \frac{s_{i,j,F}^+}{\Delta a} & \dots & 0 \end{array} \right)}_{\text{row of } \mathbf{A}_a}.$$

Finally, to solve the system, we use an implicit scheme just as in chapter 2.

Notice also that the fact that the size of  $\mathbf{A}$  is multiplicative means that the curse of dimensionality is still very much present even in continuous-time models. However, if we look at its representation in fig. 4.1 (using  $I = 40$  and  $J = 2$ ) we can see that of the  $80 \times 80$  total entries very few are non-zero (in this example less than 240). Hence, by exploiting the extreme sparsity of the system, we will be able to at least somewhat tame the issue, though problems with a large number of states will still be hard to solve.

As we did in chapter 2, it is also instructive to visualize the  $\mathbf{A}$  matrix to understand what it tells us about the solution to the problem.<sup>13</sup> First, we can immediately notice that the matrix is divided into 4 blocks, one for each possible  $(j, j')$  pair. Focusing on

---

<sup>13</sup>Since the  $\mathbf{A}$  matrix essentially contains information about the savings policy function, we could in principle look at that directly (which is arguably easier and more intuitive). However, understanding the mapping between the structure of  $\mathbf{A}$  and the savings policy function also makes debugging code much easier. This is especially useful since most of the times bugs in these models immediately reflect in the  $\mathbf{A}$  matrix.

the north-western block – associated with households who are in the low-income state,  $y_1$ , and do not transition to  $y_2$  – we immediately see that we only have elements on the main diagonal and on the first lower diagonal. This tells us that the upwind scheme is selecting the backward difference; that is, households have negative savings. This makes intuitive sense because households in the low income state expect their earnings to increase in the future and consume more to smooth consumption over time.

On the other hand, focusing on the south-eastern block, we see that things are different: in the first few rows (which are associated with lower levels of assets) we have elements on the main diagonal and on the first upper diagonal; that is, the upwind scheme selects the forward difference (i.e. households are saving). Then, for higher levels of assets we have elements on the main diagonal and on the first lower diagonal; that is, households are dissaving. Hence, this tells us that households in the higher income state,  $y_2$ , will choose to either save or dissave depending on their level of assets: wealth-poor households save to move away from the borrowing constraint until they have built a sufficiently large “buffer-stock” level of savings, while wealth-rich households dissave to raise their consumption level (because they are sufficiently far away from the borrowing constraint that their desire to increase consumption is stronger than their precautionary saving motive).<sup>14</sup>

Finally, the two blocks at north-east and south-west are simply related to the transition across income states and will all equal to  $\lambda_1$  at north-east and to  $\lambda_2$  at south-west.

### 4.4.3. Algorithm

We are at this point ready to actually solve the model. Algorithm 4 presents a standard continuous-time algorithm to solve for the stationary equilibrium of the Aiyagari model: we start by defining tolerances and initial guesses for both the value function and the interest rate, which are the two main objects we will be iterating on. Then, given our guess for the interest rate, we solve the agent’s problem by iterating on the value function until convergence.<sup>15</sup> Once the household problem has converged, we “extract” the matrix  $\mathbf{A}$  from the last HJB iteration which we can use to solve the KFE and get the stationary distribution by transposing it,  $\mathbf{A}^T$ , and solving the eigenvalue problem  $0 = \mathbf{A}^T \mathbf{g}$ . With the stationary distribution on hand, we can then easily compute capital supply and the interest rate implied by the firm’s FOC. If the latter is not sufficiently close to our guess for the interest rate, we update the interest rate accordingly and restart the algorithm.

---

<sup>14</sup>How large “buffer-stock” savings are depends not only on the borrowing constraint but, most importantly, on the earnings process itself: in the current version of the model with only two income states, households decide based on the likelihood of switching to the low-income state and on the expected duration of the latter; in more complicated models, this will naturally depend both on the volatility and on the persistence of the earnings process.

<sup>15</sup>We do not actually need a guess for the wage rate since it can be recovered directly from the firm’s FOC as a function of the interest rate

**Algorithm 4** Algorithm for the two-state Aiyagari model in continuous-time.

Define arbitrarily small tolerances  $\varepsilon_r$  and  $\varepsilon_v$

Start with initial guesses  $r^0$  and  $\mathbf{v}^0$

**for**  $n = 0, 1, 2, \dots$  **do**

    1. Given prices  $r^n$  and  $w^n = w(r^n)$ , solve the agent's problem:

**for**  $k = 0, 1, 2, \dots$  **do**

            Given a guess  $\mathbf{v}^k$ , find  $\mathbf{v}^{k+1}$  by solving the HJB:

$$\frac{\mathbf{v}^{k+1} - \mathbf{v}^k}{\Delta} + \rho \mathbf{v}^{k+1} = \mathbf{u}^k + \mathbf{A}^k \mathbf{v}^{k+1}$$

**if**  $\|\mathbf{v}^{k+1} - \mathbf{v}^k\| < \varepsilon_v$  **then**

                Exit loop

**else**

                Set  $k = k + 1$

**end if**

**end for**

    2. Compute the stationary distribution  $\mathbf{g}$  by solving the KFE:

$$0 = \mathbf{A}^T \mathbf{g}$$

    where  $\mathbf{A}$  comes from the last HJB iteration

    3. Compute aggregate capital supply  $A = \sum_j \int ag(a, y_j) da$

    4. Find the interest rate implied by the firm FOC:  $r^* = \alpha(A/L)^{\alpha-1} - \delta$

**if**  $\|r^* - r^n\| < \varepsilon_r$  **then**

            Exit

**else**

            Set  $n = n + 1$ , update  $r^n$ , and go back to step 1

**end if**

**end for**

## 4.5. Results

Having solved for the stationary equilibrium, we now turn to analyzing the main results and economic implications of the Aiyagari model.<sup>16</sup> The continuous-time formulation offers particularly sharp insights into household behavior, especially regarding the interaction between income risk, borrowing constraints, and the resulting consumption and wealth dynamics.<sup>17</sup>

The consumption policy functions displayed in fig. 4.2a reveal the usual forces at play in incomplete market models: consumption is increasing and concave in wealth for both income states, but with markedly different curvatures that reflect the underlying earnings risk.

Near the borrowing constraint, the marginal propensity to consume (MPC) is particularly high and, in fact, becomes unbounded as households approach the constraint. The continuous-time formulation also shows that households in the low income state will hit the borrowing constraint in *finite time*. Specifically, Achdou et al. (2022) shows that the saving policy function near the constraint behaves as:

$$s_1(a) \sim -\sqrt{2\nu_1(a - \underline{a})} \quad \text{as } a \rightarrow \underline{a} \quad (4.17)$$

where  $\nu_1$  captures the speed at which households approach the constraint and depends on several model parameters. The behavior of the savings policy function implies that the derivative  $s'_1(a) \rightarrow -\infty$  as  $a \rightarrow \underline{a}$ .

The intuition behind this result is the same as in discrete time: households facing borrowing constraints and income uncertainty engage in precautionary saving, but this motive weakens as wealth increases. For wealthy households, the consumption function becomes essentially linear – a result that reflects the diminishing importance of both the borrowing constraint and income uncertainty for their consumption decisions.<sup>18</sup> In fact, the saving differences between income states,  $s_2(a) > s_1(a)$  for all  $a$ , shown in fig. 4.2b reflect not just current income differences but also expectations about future income transitions. High-income households save more partly because they can afford to, but also because they rationally anticipate the possibility of transitioning to the low income state and want to smooth consumption intertemporally.

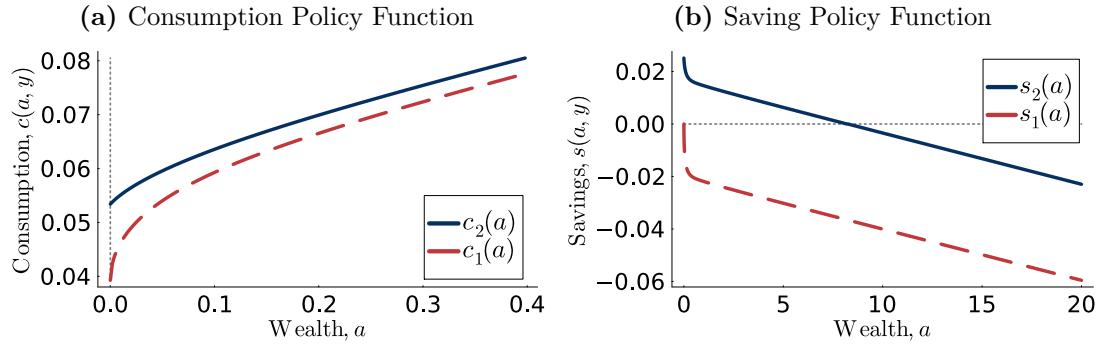
Finally, fig. 4.3 shows the stationary distribution of the Aiyagari model for each income state. The most striking feature is naturally the mass point at the borrowing constraint for low-income households, which is a direct consequence of the finite-time convergence result: households that remain in the low income state sufficiently long will eventually hit the constraint (and remain there).

---

<sup>16</sup> Appendix D.1 include julia code snippets on how to solve the HJB and KFE loops, respectively, that were used to generate all results in this chapter.

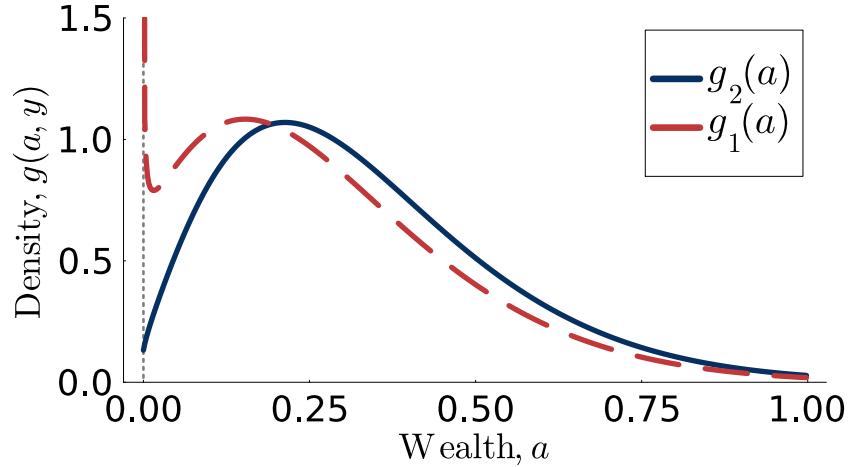
<sup>17</sup> All of the results in this section can be found almost 1-for-1 in Achdou et al. (2022).

<sup>18</sup> This asymptotic linearity, which **benhabib18:jel** show holds exactly under CRRA preferences, also helps explain why business cycle properties of heterogeneous agent models can sometimes resemble their representative agent counterparts for aggregate dynamics, an insight we will examine in more details in chapter 5.



**Figure 4.2:** Policy Functions

**Figure 4.3:** Stationary Distribution



For high-income households, the distribution is smooth and single-peaked, with an average level of wealth that is naturally higher than that of low-income households. Nonetheless, as Achdou et al. (2022) shows, the stationary distribution will be bounded above at some  $a_{\max} < \infty$  (as also implied by the savings policy function). This also means the model has a hard time explaining the high level of wealth concentration observed in the data, a well-known result in the literature.

# Chapter 5

## Aggregate Shocks

### 5.1. Introduction

Before studying heterogeneous-agent models with aggregate shocks in continuous time, it might be useful to first have a quick look at the history of modern macroeconomics as well as some of the classic results on aggregation, which will allow us to provide context to some of the results we will obtain later in this chapter.

#### 5.1.1. A Brief History of Macro<sup>1</sup>

One could say that the foundations of modern macroeconomics were laid down with the rational expectations revolution initiated by Robert J. Lucas in the 1970s. Prior to this, the field had mostly developed within the Keynesian paradigm established in the 1930s, which focused primarily on aggregate relationships with limited interest to microeconomic foundations and essentially no role for inequality or distributions.<sup>2</sup> The Lucas critique fundamentally altered this approach, demanding that macroeconomics models be grounded in explicit optimization problems of individual agents.

#### First Generation: RA Models (1970 – 1990)

The “first generation” of modern macroeconomic models, spanning from the 1970s through the early 1990s, established the representative agent (RA) framework as the dominant paradigm. This generation encompassed the Neoclassical Growth model, the Real Business Cycle model of Kydland and Prescott (1982), and the emergent New Keynesian models.

---

<sup>1</sup>It is obviously impossible to give a comprehensive summary of the history of macroeconomics in such little space. Hence, we hereby focus primarily on how the understanding and treatment of household heterogeneity evolved over time (and vastly simplify it). We mostly take the view of Kaplan and Violante (2018), while a somewhat different perspective can be found in Cherrier et al. (2023). The Macro Musings Podcast also featured an interview with Benjamin Moll on the same subject is available at <https://macromusings.libsyn.com/> – this section is also partly based on that interview.

<sup>2</sup>A few exceptions were present in growth theory, where distributions did play a role, though not quite in the same way as we would think about it today (e.g. work by Kaldor and Pasinetti on the factor income distribution or by Stiglitz and Blinder on the personal income distribution).

These models achieved analytical tractability through the assumption of a representative agent, therefore abstracting from distributional considerations by construction. While this simplification enabled significant theoretical advances in understanding business cycle dynamics and policy transmission mechanisms, it precluded any analysis of inequality or distributional effects.

### Second Generation: Early HA Models (1990 – 2008)

The limitations of representative agent models prompted the development of heterogeneous agent (HA) frameworks in the late 1980s and early 1990s. The seminal contributions in this literature emerged from the so-called Minnesota school, with the pioneering work of Bewley (1977) first, and then later Prescott's students Imrohoroglu (1989), Huggett (1993), and Aiyagari (1994). In this literature the distribution was front and center: one of its main objective was exactly to generate significant levels of consumption, income, and wealth heterogeneity that could be in line with the data.

Solving for the stationary equilibrium of these models was already challenging enough and, for reasons that will become clearer later in this chapter, it was at the time impossible to solve these models in the presence of aggregate shocks. As such, these early “second-generation” models were not at all suited for the study of business cycle fluctuations.

The computational breakthrough came with Krusell and Smith (1998), who showed that heterogeneous agent models with both idiosyncratic and aggregate shocks could be solved numerically. The key insight was the so-called “approximate aggregation”: despite substantial heterogeneity in individual wealth and income, aggregate dynamics could be accurately characterized using only limited moments of the distribution. Specifically, they showed that agents could forecast future prices with remarkable accuracy using only the mean of the wealth distribution, rather than requiring knowledge of the entire distribution.

This result had significant implications: while these models successfully generated realistic wealth and income distributions – addressing a significant empirical failure of representative agent models – their aggregate dynamics remained remarkably similar to those of representative agent economies. As Krusell and Smith (1998) demonstrated, the irrelevance of higher moments of the wealth distribution for aggregate dynamics meant that heterogeneity, while important for understanding inequality, did not substantially affect macroeconomic aggregates.<sup>3</sup> In fact, this was not just an outcome of the exact specification in Krusell and Smith (1998), but a finding that emerged persistently throughout this literature: in baseline specifications with linear decision rules and homothetic preferences, heterogeneity’s impact on aggregate dynamics remained limited. The underlying rationale is fairly straightforward: when wealthy agents simply behave as scaled versions of poor agents, aggregation remains relatively simple despite the underlying heterogeneity.

To better understand the last statement, compare the distribution of marginal propensities to consume (MPC) in the data vs. a standard incomplete market model as in fig. 5.1. In fig. 5.1a we plot the MPC along the distribution of cash on hand as measured in the

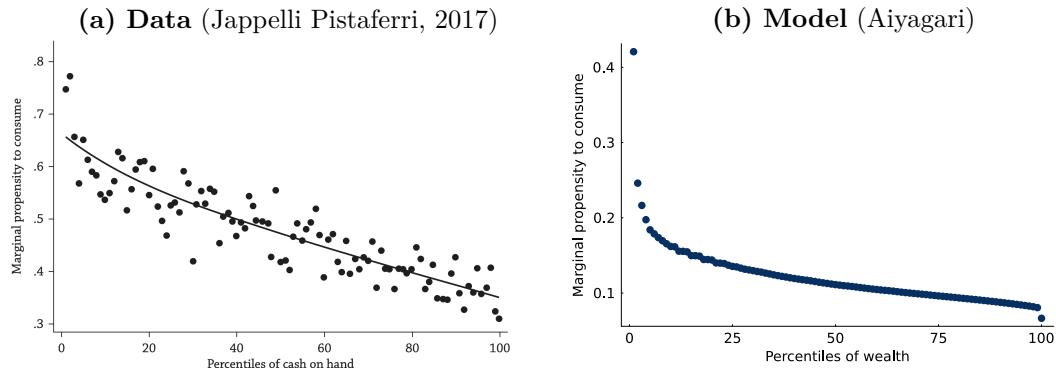
---

<sup>3</sup>As stated by Lucas (2003) “For determining the behavior of aggregates, [Krusell and Smith] discovered, realistically modeled household heterogeneity just does not matter very much. For individual behavior and welfare, of course, heterogeneity is everything.”

### 5.1. Introduction

---

data (taken from Jappelli and Pistaferri 2017), and in fig. 5.1b we plot the MPC in the Aiyagari model along the distribution of wealth. While MPCs in the data are very high (30% to 70%) and very heterogeneous, these are much lower (5% to 40%) in the model and only high very close to the borrowing constraint. As will see momentarily, this divergence between model and data is precisely the reason why in this class of models inequality does not really matter for macroeconomic aggregates.



**Figure 5.1:** MPC: data vs. incomplete market model

### Third Generation: HA Models with Aggregate Implications (2008 – .)

The Great Recession marked a significant shift in the profession’s approach to heterogeneity. The crisis highlighted phenomena – particularly the role of household balance sheets, leverage, and financial frictions – that could not be adequately addressed within either RA frameworks or the existing HA models featuring approximate aggregation.

A “third generation” of models therefore emerged in response to these empirical challenges that tried to incorporate several critical departures from earlier frameworks. First, these models take micro data substantially more seriously, calibrating not just to aggregate moments but to the entire distribution of marginal propensities to consume, wealth holdings, and portfolio compositions. Second, they incorporate non-homotheticities and nonlinearities that break the scaling properties underlying approximate aggregation. Third, they explicitly model features such as hand-to-mouth consumers, liquidity constraints, and portfolio heterogeneity that generate first-order effects on aggregate dynamics.

As emphasized by Kaplan and Violante (2018), this generation of models demonstrates that macroeconomic aggregates both affect and are affected by distributional considerations. Such connection between inequality and macroeconomic outcomes represents a fundamental departure from earlier frameworks. These models have proven essential for understanding phenomena ranging from the transmission of monetary policy to the aggregate effects of fiscal transfers (e.g., see Kaplan, Moll, et al. 2018; Kaplan, Mitman, et al. 2020; Favilukis et al. 2016, among many others).<sup>4</sup>

<sup>4</sup>The following extract from Janet Yellen speech “Macroeconomic Research After the Crisis” is a good summary of what we have been discussing in this section: “Prior to the financial crisis, representative-agent models were the dominant paradigm for analyzing many macroeconomic questions [...] A disaggregated

### 5.1.2. Aggregation

Having established the historical context for heterogeneous-agent models, we now turn to a question that is key to understanding when inequality matters for macroeconomic aggregates and provides the theoretical foundation for current models: under what conditions can we represent the behavior of a heterogeneous-agent economy through a representative agent or, alternatively, under what conditions does our economy admit aggregation? In order to answer this question, we start by defining what aggregation mean.

**Definition 12** [Aggregation]: *An economy is said to admit if the behavior of aggregate equilibrium quantities and prices does not depend on the distribution of individual quantities across agents.*

That is, when aggregation holds, we can construct a fictitious representative agent whose optimization problem yields the same aggregate dynamics as the original heterogeneous-agent economy. We distinguish three primary forms of aggregation, each with distinct requirements and implications: trivial aggregation, demand aggregation, and complete markets.

#### Trivial Aggregation

The simplest form of aggregation occurs when all agents are identical ex ante:

**Result 1** [Trivial Aggregation]: *The neoclassical model admits a representative household if:*

- i. All agents have identical preferences satisfying strict monotonicity and strict convexity
- ii. All agents have equal initial endowments

While mathematically straightforward, trivial aggregation is economically very restrictive and naturally eliminates exactly the type of heterogeneity that motivates our analysis, so we will now move to two other results that will be much more relevant for our models.

#### Demand Aggregation

A more substantive aggregation result, due to Gorman (1961), establishes conditions under which heterogeneous agents can be aggregated despite differences in wealth.

Consider an economy with  $I$  agents and  $M$  goods. Let  $c_{m,i}(\mathbf{p}, w_i)$  denote agent  $i$ 's demand for good  $m$ , where  $\mathbf{p} = \{p_m\}_{m=1}^M$  is the price vector and  $w_i$  is agent  $i$ 's wealth.

---

approach seems needed to understand some key aspects of the Great Recession. [...] While the economics profession has long been aware that these issues matter, their effects had been incorporated into macro models only to a very limited extent prior to the financial crisis. [...] I am glad to now see a greater emphasis on the possible *macroeconomic consequences of heterogeneity*" (emphasis added).

Aggregate demand for good  $m$  is then given by:

$$C_m(\mathbf{p}, w_1, w_2, \dots, w_I) = \sum_{i=1}^I c_{m,i}(\mathbf{p}, w_i)$$

The central question is: when can we express aggregate demand as a function of aggregate wealth alone (independently of its distribution)? That is, when is it the case that:

$$C_m(\mathbf{p}, w_1, w_2, \dots, w_I) = C_m\left(\mathbf{p}, \sum_{i=1}^I w_i\right) \quad (5.1)$$

For this representation to hold, wealth reallocations between agents must not affect aggregate demand. Formally, we require:<sup>5</sup>

$$\frac{\partial c_{m,i}(\mathbf{p}, w_i)}{\partial w_i} = \frac{\partial c_{m,j}(\mathbf{p}, w_j)}{\partial w_j} \Big|_{dw_i = -dw_j} \quad \forall m, i, j,$$

which is satisfied when all agents have identical marginal propensities to consume (MPC) or, equivalently, if we can write consumption as an affine function of wealth (i.e. households have linear Engel curves):

$$c_{m,i}(\mathbf{p}, w_i) = \alpha_{m,i}(\mathbf{p}) + \beta_m(\mathbf{p})w_i \quad (5.2)$$

where crucially, the coefficient  $\beta_m(p)$  is common across agents. In fact, eq. (5.2) immediately implies that we can also write:

$$\begin{aligned} C_m(\mathbf{p}, w_1, w_2, \dots, w_I) &= \sum_{i=1}^I \alpha_{m,i}(\mathbf{p}) + \beta_m(\mathbf{p})w_i \\ &= \sum_{i=1}^I \alpha_{m,i}(\mathbf{p}) + \beta_m(\mathbf{p})W = C_m(\mathbf{p}, W) \end{aligned}$$

where  $W = \sum_{i=1}^I w_i$ , which is exactly eq. (5.1).

This result was stated in terms of indirect utility by Gorman (1961) as follows:

**Theorem 5** [Gorman 1961]: *If (and only if) agents' indirect utility functions can be represented as:*

$$v_i(\mathbf{p}, w_i) = a_i(\mathbf{p}) + b(\mathbf{p})w_i$$

*then aggregate consumption can be expressed as the choice of a representative household with indirect utility*

$$v(\mathbf{p}, W) = \bar{a}(\mathbf{p}) + b(\mathbf{p})W$$

---

<sup>5</sup>Note that we can directly write  $\partial c_{m,i}(\cdot)$  instead of  $\partial C_m(\cdot)$  simply because:

$$\frac{\partial C_m(\mathbf{p}, w_1, \dots, w_I)}{\partial w_i} = \frac{\partial c_{m,i}(\mathbf{p}, w_i)}{\partial w_i} \quad \forall m, i, j.$$

where  $\bar{a}(\mathbf{p}) = \sum_i a_i(\mathbf{p})$  and  $W = \sum_i w_i$

This can be easily proven using Roy's identity, which establishes that agent  $i$ 's demand for good  $m$  is:

$$c_{m,i}(\mathbf{p}, w_i) = -\frac{\partial v_i(\mathbf{p}, w_i)/\partial p_m}{\partial v_i(\mathbf{p}, w_i)/\partial w_i}.$$

With  $v_i(\mathbf{p}, w_i) = a_i(\mathbf{p}) + b(\mathbf{p})w_i$ :

$$c_{m,i}(\mathbf{p}, w_i) = -\frac{\frac{\partial a_i(\mathbf{p})}{\partial p_m} + w_i \frac{\partial b(\mathbf{p})}{\partial p_m}}{b(\mathbf{p})} = \alpha_{m,i}(\mathbf{p}) + \beta_m(\mathbf{p})w_i$$

where  $\alpha_{m,i}(\mathbf{p}) = -\frac{\partial a_i(\mathbf{p})/\partial p_m}{b(\mathbf{p})}$  and  $\beta_m(\mathbf{p}) = -\frac{\partial b(\mathbf{p})/\partial p_m}{b(\mathbf{p})}$ . Since  $\beta_m(\mathbf{p})$  is independent of  $i$ , all agents have the same marginal propensity to consume, establishing aggregation.

Relative to the previous result about trivial aggregation, this is obviously a much more general result: we do not require everyone to be identical, but only their MPCs. Two important classes of preferences that satisfy Gorman aggregation are quasilinear utility and homothetic preferences.

**Quasilinear Utility.** — Consider preferences  $u_i(c_{1,i}) + \beta c_{2,i}$  with good 2 as numeraire subject to budget constraint  $p_1 c_{1,i} + c_{2,i} = w_i$ . The solution is therefore:

$$\begin{aligned} c_{1,i} &= (u'_i)^{-1}(\beta p_1) \\ c_{2,i} &= w_i - p_1(u'_i)^{-1}(\beta p_1) \end{aligned}$$

Both demands exhibit constant marginal propensities out of wealth (zero and one, respectively), satisfying Gorman aggregation. Note that the indirect utility function becomes:

$$\begin{aligned} v_i(p_1; w_i) &= u_i((u'_i)^{-1}(\beta p_1)) + \beta [w_i - p_1(u'_i)^{-1}(\beta p_1)] \\ &= \alpha_i(p_1) + \beta w_i \end{aligned}$$

which is linear in individual wealth and has common coefficient  $\beta$ , so we show again that we satisfy Gorman aggregation.

**Homothetic Preferences.** — With homothetic utility, the ratio of goods demanded depends only on relative prices, not on wealth levels. This property ensures that Engel curves are linear through the origin, another special case of Gorman aggregation.

## Complete Markets Aggregation

The third aggregation result applies when markets are complete, allowing agents to insure against all idiosyncratic risks. This approach was introduced by Negishi (1960) – which suggested a method to calculate the competitive equilibrium (CE) of complete market economies with heterogeneous households – and was later generalized by Constan-

## 5.1. Introduction

---

tinides (1982) and refined by Ogaki (2003).<sup>6</sup> This approach is particularly useful for those economies where Gorman aggregation does not apply and we therefore do not know how to write the preferences of the representative agent.<sup>7</sup>

**The Negishi Approach** — The key insight is that any CE in a complete markets economy corresponds to the solution of a social planner's problem with appropriately chosen Pareto weights. The challenge is finding the weights that correspond to our desired CE in the original economy.

Consider an economy with heterogeneous agents indexed by  $i \in \{1, \dots, I\}$  with initial wealth shares  $\{\omega_{i0}\}_{i=1}^I$ . The Negishi planner's problem is:

$$\max_{\{c_{it}\}} \sum_{t=0}^{\infty} \beta^t \sum_{i=1}^I \alpha_i u(c_{it}) \quad (5.3)$$

subject to the aggregate resource constraint, where  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_I)$  are Pareto weights satisfying  $\sum_i \alpha_i = 1$ . The Negishi method then boils down to guessing a vector of weights (as opposed to the entire price sequence) and ensuring that the time-zero Arrow-Debreu budget constraint of each agent holds exactly at the guessed vector of weights.<sup>8</sup>

**The Constantinides Decomposition** — We just learned that the equilibrium allocations of a complete market economy with heterogeneity can be obtained as the solution of a Negishi planner problem. We can then use this approach to prove a more general aggregation result for complete market economies. Essentially, we split the problem in two stages: in the first stage, given aggregate consumption, we solve the problem of allocating consumption across agents and then, in the second stage, we solve the problem of choosing the optimal path of aggregate consumption.

1. **Allocation Stage:** Given aggregate consumption  $C_t$ , solve for the optimal allocation across agents:

$$U(C_t) = \max_{\{c_{it}\}} \sum_{i=1}^I \alpha_i u(c_{it}) \quad \text{subject to} \quad \sum_{i=1}^I c_{it} = C_t \quad (5.4)$$

This yields an indirect utility function  $U(C_t)$  for the planner.

2. **Aggregation Stage:** Choose the optimal aggregate consumption path:

$$\max_{\{C_t\}} \sum_{t=0}^{\infty} \beta^t U(C_t) \quad (5.5)$$

---

<sup>6</sup>See, L. Maliar and S. Maliar (2003) and L. Maliar and S. Maliar (2001) for some examples.

<sup>7</sup>A thorough demonstration of why and how complete markets allow aggregation is beyond the scope of these notes, we refer the interested reader to Ogaki (2003) as well as classic macro textbook references such as Ljungqvist and Sargent (2018).

<sup>8</sup>Incidentally, it turns out that the Pareto weights in the social planner's problem are determined by the initial wealth distribution in the competitive equilibrium. Specifically, agents with higher initial wealth will receive higher weights in the planner's problem.

subject to the aggregate resource constraint.

Importantly, this approach shows that the aggregate dynamics of the model can be described by the solution to the problem of a representative agent (RA) – whose preferences are, however, different from the preferences of individual consumer (i.e.  $U \neq u$ ), which will generally depend on the initial wealth distribution through the Pareto weights.

The Constantinides aggregation theorem is a very general result on the existence of a representative agent: the only restriction on preferences is strict concavity of  $u$ , but neither homotheticity nor quasilinearity are required. While the restrictions on preferences are minimal, however, it does require complete markets – which demand aggregation does not.

Concluding, we argued that we can represent the aggregate economic dynamics of a complete-market economy through the problem faced by a representative agent and, under complete markets – where a competitive equilibrium is Pareto optimal – the RA’s problem in turn can be framed as the Social Planner’s problem. This equivalence shows that under rather general conditions, if markets are complete, aggregation is possible.

## 5.2. Aggregate Shocks in Continuous Time

Having established the foundations of HA models in stationary environments in chapter 4, we now extend our analysis to incorporate aggregate uncertainty. Under the assumption of rational expectations adding aggregate uncertainty in a heterogeneous-agent framework means that, in order to correctly forecast the evolution of prices, households must track how the entire cross-sectional distribution evolves stochastically over time. As we will see momentarily, this is an infinitely more complicated problem to solve; so complicated, in fact, that we will not actually be able solve it and will have to bypass one way or another.

### 5.2.1. General Framework

Consider a continuum of agents  $\mathcal{I} = [0, 1]$ , where each agent  $i \in \mathcal{I}$  has a state variable  $X_t^i \in \mathbb{R}$  that evolves according to the following SDE:

$$dX_t^i = \mu_t(X_t^i)dt + \sigma_t(X_t^i)dW_t^i + \underbrace{\sigma_t^0(X_t^i)dW_t^0}_{\text{Common Noise}} \quad (5.6)$$

where  $W_t^i$  represents an independent one-dimensional Brownian motion specific to agent  $i$ , and  $W_t^0$  denotes a one-dimensional Brownian motion *common to all agents*. The presence of this common noise term  $dW_t^0$  introduces aggregate uncertainty that affects all agents simultaneously.

Let  $g_t(x)$  denote the population density across state  $x$  at time  $t$ . The following proposition establishes how the distribution evolves in the presence of aggregate shocks:<sup>9</sup>

---

<sup>9</sup>The proof is left as an exercise and follows the same steps as the derivation of the KFE without common noise in appendix C.1, with the crucial distinction that the law of large numbers does not apply to  $W_t^0$  (i.e. the common noise term does not vanish under aggregation).

**Proposition 3** [KFE with Common Noise]: *Suppose that each agent's state,  $X_t^i$ , evolves according to:*

$$dX_t^i = \mu_t(X_t^i)dt + \sigma_t(X_t^i)dW_t^i + \sigma_t^0(X_t^i)dW_t^0.$$

*Then the population density,  $g_t$ , satisfies the following Kolmogorov Forward equation:*

$$dg_t(x) = \left\{ -\partial_x [\mu_t(x)g_t(x)] + \frac{1}{2}\partial_{xx} [(\sigma_t^2(x) + (\sigma_t^0)^2(x)) g_t(x)] \right\} dt - \partial_x [\sigma_t^0(x)g_t(x)] dW_t^0 \quad (5.7)$$

Relative to the case without common noise analyzed in chapter 4, the structure of eq. (5.7) merits additional examination. The drift component:

$$\mu_{g,t} = -\partial_x [\mu_t(x)g_t(x)] + \frac{1}{2}\partial_{xx} [(\sigma_t^2(x) + (\sigma_t^0)^2(x)) g_t(x)]$$

contains the standard advection and diffusion terms familiar from the deterministic case, and the additional variance term  $(\sigma_t^0)^2(x)$  in the diffusion coefficient represents the standard Itô correction that now also arises from the common noise. The volatility component:

$$\sigma_{g,t} = -\partial_x [\sigma_t^0(x)g_t(x)],$$

which instead was not present in the absence of common noise, captures how the entire distribution shifts in response to aggregate shocks. Because  $dW_t^0$  affects all agents simultaneously, the distribution  $g_t(x)$  now becomes a stochastic object whose evolution depends on the realization of aggregate uncertainty. In fact, the KF equation (5.7) is now a *stochastic PDE*.

### 5.2.2. The Problem with Rational Expectations

Under rational expectations households know that prices are equilibrium objects and that, as such, they depend on the cross-sectional distribution. Hence, in order to take rational decisions, households would need to have expectations about the evolution of prices that are consistent with their actual evolution – which means that, in order to forecast future prices, household would also have to correctly forecast the evolution of the distribution. This would require the cross-sectional distribution, an infinite-dimensional object, to enter as a state variable in the individual optimization problem. In the presence of heterogeneity, globally solving the household problem under rational expectations is therefore impossible.<sup>10,11</sup>

The literature has developed several approaches to “approximate” rational expectations equilibria in this setting. We categorize these methods based on their underlying

---

<sup>10</sup>In fact, also in the standard incomplete markets models without aggregate uncertainty, prices come from market clearing and therefore depend on the entire cross-sectional distribution. Even in those models, households need to know the evolution of the distribution to forecast prices; however, in those models we can usually ignore the issue by focusing on stationary equilibria in which the distribution (and therefore prices) is invariant.

<sup>11</sup>See Moll (2024) for a more detailed discussion of the causes (and consequences) of modeling rational expectations in HA models, or Moll and Ryzhik (2025) for a complementary discussion of the same issue in the context of mean field games.

approximation strategy:<sup>12</sup>

**Statistical Approximation Methods** — These methods usually replace the infinite-dimensional distribution with a finite set of sufficient statistics that households must track to forecast prices. The pioneering work of Krusell and Smith (1998) demonstrated that low-order moments often provide adequate approximation. Recent advances in the same space also employ machine learning techniques, as in Fernández-Villaverde et al. (2023), to capture more complex relationships between distributional statistics and equilibrium prices.

**Perturbation Methods** — Perturbation approaches usually linearize the model around either a deterministic steady state (Reiter 2009; Ahn et al. 2017) or a stochastic steady state (**bhandari23:wp**). These methods exploit the local structure of the model to construct Taylor approximations of the equilibrium dynamics. The key insight is that for sufficiently small aggregate shocks, the distribution evolves approximately linearly around its steady-state value.

**Master Equation Approach** — Recent work by Bilal (2023) and Gu et al. (2023) develops methods based on the master equation from mean field game theory. This approach characterizes the equilibrium through a partial differential equation in the space of distributions – the master equation – which encodes both how the value function depends on the entire distribution (from the HJB equation) and the evolution of the distribution (from the KFE). The master equation therefore includes all relevant pieces of information about our economy and, if we could find a solution to it, we would be able to find the equilibrium of our economy. The approaches in Bilal (2023) and Gu et al. (2023) boil down to proposing different ways of approximating the solution to the master equation.<sup>13,14</sup>

### 5.2.3. Krusell and Smith

We now examine how to solve a model with aggregate shocks using the regression approach introduced by Krusell and Smith (1998). While more recent solution methods have been developed, the model is a natural extension of the Aiyagari model studied in chapter 4 and transparently illustrates the phenomenon of approximate aggregation. Setting up the

---

<sup>12</sup>A full treatment of all these different methods goes well beyond the scope of these notes, so we hereby just briefly mention the most commonly used ones and *some* of the key references.

<sup>13</sup>The exact specification of the master equation clearly depends on the model being considered but, in its most basic form, it looks something like:

$$\rho v(x, g) = \max_c u(c) + \mathcal{A}(x, g)[v] + \left\langle \frac{\delta v(x, g)}{\delta g}, \mathcal{A}^*[g] \right\rangle \quad (5.8)$$

where  $x$  is a vector of (finite-dimensional) state variables,  $\mathcal{A}$  and  $\mathcal{A}^*$  are the usual HJB and KF operators,  $\delta v / \delta g$  is the infinite-dimensional Fréchet derivative, and  $\langle \cdot \rangle$  is the inner product on the space of functions.

<sup>14</sup>While there is plenty of resources online on the master equation, much of it might be overwhelming at this stage. I would therefore suggest to start from Bilal (2023, especially sections 1 and 2), then move to Cardaliaguet et al. (2019), and only then to the more complete treatment in Carmona and Delarue (2018).

model in continuous time obviously requires some departures from the original discrete-time formulation; we will emphasize the differences as they show up.

## Model Setup

On the household side, the structure mirrors exactly that of the Aiyagari model examined in section 4.3: the economy populated by a continuum of households indexed by  $i \in [0, 1]$  who face both idiosyncratic and aggregate uncertainty. Each household  $i$  has preferences over consumption streams given by:

$$\mathbb{E}_0 \int_0^\infty e^{-\rho t} u(c_i(t)) dt \quad (5.9)$$

where  $\rho > 0$  denotes the discount rate and  $u(\cdot)$  satisfies standard regularity conditions. Each household supplies labor inelastically and faces idiosyncratic income risk through a two-state Poisson process  $y \in \{y_L, y_H\}$  with transition intensities  $\lambda(y)$ . Households can save in a risk-free asset subject to a borrowing constraint  $a \geq \underline{a}$ .

The production side introduces aggregate uncertainty through a stochastic total factor productivity process. Specifically, output is produced according to:

$$Y_t = e^{z_t} K_t^\alpha L_t^{1-\alpha}$$

where  $Z_t = e^{z_t}$  represents total factor productivity and  $z_t$  follows an Ornstein-Uhlenbeck process:

$$dz_t = \eta_z (\bar{z} - z_t) dt + \sigma_z dW_t \quad (5.10)$$

where  $dW_t$  denotes an *aggregate* Brownian shock common to all agents, and we denote by  $\mathcal{L}$  the infinitesimal generator of the process in eq. (5.10):

$$(\mathcal{L}f)(z) = \eta_z (\bar{z} - z) f'(z) + \frac{\sigma_z^2}{2} f''(z) \quad (5.11)$$

This formulation differs from Krusell and Smith (1998) in two respects: First, we model TFP as a continuous process rather than a two-state Markov chain. Second, we assume that idiosyncratic and aggregate shocks are independent, whereas the original paper allows for correlation through state-dependent transition probabilities.

Factor prices satisfy the usual first-order conditions of the firm:

$$r_t = \alpha e^{z_t} \left( \frac{K_t}{L_t} \right)^{\alpha-1} - \delta \quad (5.12)$$

$$w_t = (1 - \alpha) e^{z_t} \left( \frac{K_t}{L_t} \right)^\alpha \quad (5.13)$$

Crucially, current prices depend on current aggregate productivity  $z_t$  and, through aggregate capital, on the entire distribution of wealth holdings:  $K_t = \int adG_t$ .

Equations (5.12) and (5.13) make apparent the problem of solving the rational expectations model: to form expectations about the evolution of prices, households must predict

the evolution of aggregate capital, which depends on the savings decisions of all agents (through market clearing) and thus on the entire cross-sectional distribution  $g_t(a, y)$ . We now formalize this problem and introduce the key approximation that makes it tractable.<sup>15</sup>

### The Infinite-Dimensional Problem

To understand the computational challenge, let us first write the HJB equation for the full rational-expectations problem:

$$\rho v = \max_c u(c) + \mathcal{A}_c v + \mathcal{Q}v + \mathcal{L}v + \frac{\delta v}{\delta g} \partial_t g \quad (5.14)$$

where the operators  $\mathcal{A}_c$  and  $\mathcal{Q}$  are the generator of the wealth and earnings processes as in chapter 4 and are given by:

$$(\mathcal{Q}f)(y) = \mathbf{1}_{\{y=y_1\}} \lambda(y_1)[f(y_2) - f(y_1)] + \mathbf{1}_{\{y=y_2\}} \lambda(y_2)[f(y_1) - f(y_2)] \quad (5.15)$$

$$(\mathcal{A}_c v)(a, y; z, g) = [w(z, g)y + r(z, g)a - c]v_a(a, y; z, g) \quad (5.16)$$

and, crucially, the term  $\frac{\delta v}{\delta g} \partial_t g$  involves a functional derivative of the value function with respect to the distribution.

To understand why this term is problematic, recall that  $g_t$  is an infinite-dimensional object – it is a function that specifies the mass of agents at each possible  $(a, y)$  pair. The functional derivative  $\frac{\delta v}{\delta g}$  represents how the value function changes in response to perturbations of the distribution at each point in the state space. In particular, when taking the derivative of the  $v$  functional with respect to the infinite-dimensional distribution  $g$ ,  $\frac{\delta v}{\delta g}$ , the correct notion of derivative is that of the Fréchet derivative. Specifically, consider how the value function  $v$  responds to small changes in the distribution  $g$ . If the household state space were discrete with only finitely many possible states, then the distribution would be characterized by a vector of masses  $(g_1, g_2, \dots, g_N)$  at each state, and the distributional derivative would be given by the vector of standard partial derivatives  $\left( \frac{\partial v}{\partial g_i} \right)_{i=1}^N$  measuring how the value function changes when the mass at state  $i$  increases or decreases.

The Fréchet derivative simply extends this concept to continuous state spaces. For the functional  $v(\cdot, g)$ , the Fréchet derivative  $\frac{\delta v}{\delta g}(a, y; z, g)[a', y']$  measures how the value function responds to adding an infinitesimal mass of households at the specific location  $(a', y')$  in the wealth-income space.<sup>16</sup> Standard numerical methods cannot handle such infinite-dimensional operations, so one generally has to find a way around it.

---

<sup>15</sup> Households also obviously need to correctly forecast the evolution of the exogenous aggregate state  $z_t$  but that is easily accounted for since its evolution in eq. (5.10) is common knowledge.

<sup>16</sup> The notation  $\frac{\delta v}{\delta g}(a, y; z, g)[a', y']$  simply emphasizes that this derivative depends on both the current state  $(a, y; z, g)$  where we evaluate the functional and the specific location  $(a', y')$  where we consider the perturbation – just as a finite-dimensional gradient depends on both the evaluation point and the direction of the perturbation.

## Bounded Rationality

The crucial insight of Krusell and Smith (1998) is to assume that agents only use a finite-dimensional approximation of the distribution when forming expectations. Specifically, agents are **boundedly rational** in the sense that they base their decisions on a finite set of moments of the distribution rather than the entire distribution itself.

In particular, agents are assumed to believe that prices only depend on the first  $I$  moments of the distribution, with  $I < \infty$ . Let  $\mathbf{m} = (m_1, m_2, \dots, m_I)$  denote the vector of the first  $I$  moments of the distribution – for example,  $m_1$  might represent mean wealth,  $m_2$  the variance of wealth, and so forth. Under the bounded rationality assumption laid out in Krusell and Smith (1998), agents believe that these moments evolve according to a **perceived law of motion** (PLM):

$$\mathbf{m}' = h_I(\mathbf{m}, z)$$

or, equivalently in its continuous time formulation:

$$d\mathbf{m} = h_I(\mathbf{m}, z)dt \quad (5.17)$$

where  $h^I$  is an arbitrarily defined function mapping current moments and productivity states to future moments.

Under this approximation, the value function simplifies to  $v(a, y; z, \mathbf{m})$ , which only depends on a finite set of moments of  $g$  rather than on the full distribution  $g$  itself. The HJB equation then becomes:

$$\rho v(a, y; z, \mathbf{m}) = \max_c \{u(c) + \mathcal{A}_c v + \mathcal{L}v + \nabla_{\mathbf{m}} v \cdot h_I(\mathbf{m}, z)\} \quad (5.18)$$

where we have used the notation  $\nabla_{\mathbf{m}} v$  to indicate the gradient of  $v$  with respect to the vector of moments.

This formulation transforms the infinite-dimensional problem into a tractable finite-dimensional one where the number of moments  $I$  effectively represents the degree of sophistication in agents' expectations. As we will see, even with  $I = 1$  (using only mean capital), the model can achieve remarkable accuracy in many cases.<sup>17</sup>

## HJB and KF Equations

We now derive the complete system of equations characterizing the bounded-rationality equilibrium. For concreteness, we follow the original paper and assume agents track only the mean of the wealth distribution, setting  $I = 1$  and  $m = K$  (aggregate capital). In

---

<sup>17</sup>Notice that, by solving this model, we are “only” solving and finding a bounded-rationality equilibrium of the Krusell-Smith economy, and not necessarily approximating the rational-expectations equilibrium. If we wanted to state that the bounded-rationality equilibrium we obtain is also an approximation of the rational-expectations equilibrium, then we would also need to show that this is indeed the case.

their discrete-time formulation, Krusell and Smith (1998) postulate the following PLM:

$$\log K' = a_0 + a_1 \log K \quad \text{if } Z = Z_g \quad (5.19)$$

$$\log K' = b_0 + b_1 \log K \quad \text{if } Z = Z_b \quad (5.20)$$

That is, capital tomorrow is a log-linear function of capital today with coefficients that are allowed to change based on the state of the economy.<sup>18</sup>

In continuous time we assume that agents believe aggregate capital to evolve according to:

$$d \log K_t = (\theta_0 + \theta_1 z_t + \theta_2 \log K_t + \theta_3 z_t \log K_t) dt.$$

Applying Itô's lemma with  $X_t = \log K_t$  and  $f(X_t) = e^{X_t}$ , we obtain:

$$dK_t = K_t(\theta_0 + \theta_1 z_t + \theta_2 \log K_t + \theta_3 z_t \log K_t) dt \equiv h(K_t, z_t) dt$$

where

$$h(K_t, z_t) = \frac{\mathbb{E}[dK_t | K_t, z_t]}{dt}.$$

The HJB equation for the household problem then becomes:

$$\rho v(a, y; z, K) = \max_c u(c) + \mathcal{A}_c v + \mathcal{Q}v + \mathcal{L}v + \frac{\partial v}{\partial K} h(K, z) \quad (5.21)$$

and, just as in eq. (5.18), getting rid of the direct dependency of  $v$  on  $g$  allows us to solve eq. (5.21) using our usual finite-differences algorithms.

The corresponding KF equation governs the evolution of the distribution:

$$\partial_t g_t(a, y) = -\partial_a [s(a, y, r_t, w_t) g_t(a, y)] + \lambda(\tilde{y}) g_t(a, \tilde{y}) - \lambda(y) g_t(a, y). \quad (5.22)$$

Note that, unlike eq. (5.7), eq. (5.22) does not directly feature a common noise term. The reason is that, although the model feature aggregate uncertainty, the interest rate is instantaneously riskless (and therefore so are savings): the aggregate shock term does not *directly* affect the evolution of households state variables. To understand this, it is useful to examine the nature of uncertainty in the Krusell and Smith model. Consider the discrete-time limit as  $\Delta t \rightarrow 0$ . At time  $t$ , the interest rate  $r_t$  is entirely determined by current capital  $K_t$  and productivity  $Z_t$ . This rate applies to savings over the interval  $[t, t + \Delta t]$  and is predetermined with respect to the shock realization at  $t + \Delta t$ .

Recall that individual wealth dynamics are given by:

$$dat = [w_t y + r_t a_t - c_t] dt$$

Formally,  $r_t$  and  $w_t$  are  $\mathcal{F}_t$ -measurable (i.e. known at time  $t$ ), where  $\mathcal{F}_t$  denotes the

---

<sup>18</sup>Equivalently, we can also rewrite eqs. (5.19) and (5.20) in a single equation as:

$$\log K' = a_0 + (b_0 - a_0) \mathbf{1}_{Z_b} + a_1 \log K + (b_1 - a_1) \mathbf{1}_{Z_b} \log K.$$

information filtration. The aggregate shock  $dW_t$  affects future prices, but does not directly enter the wealth accumulation equation. If we were to derive the KFE from first principles, we would find that the generator for the wealth process contains no second-order terms in  $a$ , as the individual wealth dynamics have no diffusion component. This is in contrast with models featuring risky asset returns, where wealth dynamics would include a diffusion term (and so would the KFE). Notice however that, although the KFE itself contains no explicit stochastic terms, the distribution  $g_t$  remains stochastic through its dependence on the history of aggregate shocks via the path of prices. Effectively, because it only depends on  $(K_t, z_t)$  and not on  $dW_t$ , at time  $t$  the evolution of prices in an interval of length  $dt$  is known. Once households have made their decisions based on  $r_t$  and  $w_t$ , the evolution of wealth between  $t$  and  $t + dt$  becomes deterministic: it does not depend on “future” shocks  $dW_t$ . Therefore also the evolution of the entire wealth distribution between  $t$  and  $t + dt$  is itself deterministic and the KFE does not feature a term depending on  $dW_t$ . Nonetheless, as  $dW_t$  will affect prices at  $t + dt$ , the cross-sectional distribution still moves stochastically over time.<sup>19</sup>

This concludes the description of the bounded-rationality equilibrium. The solution algorithm, which we detail next, then involves finding parameters  $\Theta = (\theta_0, \theta_1, \theta_2, \theta_3)$  such that the perceived law of motion is consistent with the actual law of motion generated by aggregating individual policies.

## Solution Method

To conclude this section, we present the complete algorithm for solving the Krusell-Smith model in continuous time. The algorithm implements the bounded rationality approach, wherein agents forecast future prices using a finite set of moments rather than the entire wealth distribution.

**Implementation Considerations** — Several technical aspects require careful attention when implementing algorithm 5:

First, it is important that – once a sequence of aggregate shocks  $\{dW_t\}_{t=0}^T$  has been generated in Step 1 – one always uses the same sequence of exogenous aggregate states  $\{z_t\}_{t=0}^T$  throughout all iterations of the algorithm. Changing the sequence of shocks across iteration would introduce additional sampling noise, potentially preventing convergence on the perceived law of motion parameters  $\Theta$ .

Second, correctly writing down the discretization of the infinitesimal generator  $\mathcal{A}$  in Step 2.b is certainly the most involved part of the entire algorithm. Fortunately, the linearity of infinitesimal generators allows us to use the following decomposition:

$$\mathcal{A} = \mathcal{A}_c + \mathcal{Q} + \mathcal{L} + \mathcal{A}_K$$

---

<sup>19</sup>In this sense, the KF eq. (5.22) defines the evolution of the cross-sectional distribution *conditional* on the realization of  $K_t$  and  $Z_t$  (through market-clearing prices  $r_t = r(K_t, Z_t)$  and  $w_t = w(K_t, Z_t)$ ). One could in principle also write an *unconditional* KF equation. However, that would only add unnecessary complications at this stage.

**Algorithm 5** Krusell and Smith model in continuous-time.

1. Generate a sequence of aggregate shocks  $\{Z_t\}_{t=0}^T$  from the specified stochastic process
  2. Select the number of moments  $I$  and specify the functional form for the perceived law of motion  $h_I$
- while** true **do**

- 2.a. Guess vector of coefficients  $\Theta^n$  for the law of motion  $h_I^n(K, z)$
- 2.b. Given a law of motion  $h_I^n$ , solve the agents' problem by iterating on the HJB eq. (5.21):

$$\rho v = \max_c u(c) + \mathcal{A}v$$

where  $\mathcal{A}$  denotes the infinitesimal generator incorporating both individual and aggregate state dynamics

- 2.c. Given  $\mathcal{A}$ , compute the simulated sequence of aggregate capital  $\{K_t\}_{t=0}^T$  by solving the Kolmogorov Forward equation eq. (5.22):

$$\partial_t g_t(a, y) = \mathcal{A}_t^* g_t(a, y)$$

where market clearing requires  $K_t = \int a g_t(a, y) da$  at each time  $t$

- 2.d. Estimate new coefficients  $\Phi^n = \{\phi_j\}_{j=0}^3$  for the perceived law of motion using OLS:

$$\frac{\Delta \log K_{t+\Delta}}{\Delta t} = \phi_0 + \phi_1 z_t + \phi_2 \log K_t + \phi_3 z_t \log K_t$$

- if**  $\|\Theta^n - \Phi^n\|$  is small enough **then**  
     Exit inner loop  
**else**  
     Update  $n \leftarrow n + 1$ , set  $\Theta^{n+1} = \nu \Theta^n + (1 - \nu) \Phi^n$ , return to Step 2b  
**end if**  
**end while**
3. Measure the model's goodness of fit (e.g. by computing the  $R^2$  in step 2.d.)
- if** goodness of fit is high enough **then**  
     Accept solution  
**else**  
     Modify  $I$  or functional form  $h_I$ , return to Step 2.a.  
**end if**

where  $(\mathcal{A}_K f)(K, z) = h_I(K, z)f_K(K, z)$ . Notice that, while the finite difference terms in  $\mathbf{A}_c$  will depend on all states (in the KF eq. (5.22) savings are given by  $s(a, y, r_t, w_t)$ ), all others are independent of wealth:  $\mathcal{L}$  and  $\mathcal{Q}$  can even be discretized outside of the main loop. The finite-difference representation of  $\mathcal{A}$ , which will be of size  $N \times N$  with  $N = N_a \cdot N_y \cdot N_z \cdot N_K$  can therefore be written as:

$$\mathbf{A} = \mathbf{A}_c + \mathbf{Q}^N + \mathbf{L}^N + \mathbf{A}_K^N$$

where  $\mathbf{A}_c$  is, by construction, already of size  $N \times N$ ; while the finite-difference representations of  $\mathcal{Q}$ ,  $\mathcal{L}$ , and  $\mathcal{A}_K$  are respectively:  $\mathbf{Q}$ , of size  $N_y \times N_y$ ,  $\mathbf{L}$ , of size  $N_z \times N_z$ , and  $\mathbf{A}_K$ , of size  $(N_z \cdot N_K) \times (N_z \cdot N_K)$ . Hence, these need to be correctly “recasted” in matrices of size  $N \times N$  as follows:

$$\begin{aligned}\mathbf{Q}^N &= \mathbf{I}_{N_K} \otimes \mathbf{I}_{N_z} \otimes \mathbf{Q} \otimes \mathbf{I}_{N_a} \\ \mathbf{L}^N &= \mathbf{I}_{N_K} \otimes \mathbf{L} \otimes \mathbf{I}_{N_y} \otimes \mathbf{I}_{N_a} \\ \mathbf{A}_K^N &= \mathbf{A}_K \otimes \mathbf{I}_{N_y} \otimes \mathbf{I}_{N_a}.\end{aligned}$$

Third, notice that the distribution  $g_t$  is naturally a function of the idiosyncratic states  $(a, y)$ , while its dependence on the aggregate states  $(K, z)$  manifests indirectly through time. Consequently, unlike  $\mathbf{A}$ , the discretization of the generator  $\mathcal{A}_t^*$  in the KFE will be of size  $(N_a \cdot N_y) \times (N_a \cdot N_y)$ , yet varies at each time step due to the evolution of the aggregate state.<sup>20</sup> Hence, in order to solve the KFE, at every time step  $t$  we will need to “select” the right submatrix of  $\mathbf{A}$  based on the realization of the aggregate states  $(K_t, z_t)$ . In practice, because we approximate  $K_t$  and  $z_t$  on a grid, their realizations might not fall *on* the grid and one needs to take a stance on how to select or approximate the “correct”  $\mathbf{A}_t^T$ . One way to solve this is to simply use bilinear interpolation on the closest grid points in each dimension: we start by calculating the two closest points to  $z_t$  on its grid, denote them by  $z_t^d$  and  $z_t^u$ , and the corresponding distance  $\omega_{z_t} = z_t - z_t^d$  as well as for  $K_t$ ; then simulate the KFE for each of the four possible combinations using the corresponding submatrix and weight them accordingly. For example, to compute the distribution corresponding to the pair  $(z_t^d, K_t^d)$  we use the usual implicit method to find  $\mathbf{g}_{t+\Delta_t}^{dd}$ :<sup>21</sup>

$$\frac{\mathbf{g}_{t+\Delta_t}^{dd} - \mathbf{g}_t}{\Delta_t} = \mathbf{A}_{dd}^T \mathbf{g}_{t+\Delta_t}^{dd},$$

where  $\mathbf{A}_{dd}^T = \mathbf{A}_{c,dd} + \mathbf{Q} \otimes \mathbf{I}_{N_a}$  and  $\mathbf{A}_{c,dd}$  is the sub-block of  $\mathbf{A}_c$  corresponding to  $(z_t^d, K_t^d)$ . Once all four hypothetical distributions have been computed, we can use bilinear interpolation.

---

<sup>20</sup>Notice in fact that the infinitesimal generator in the HJB  $\mathcal{A}$  does not depend on time – and consequently neither does its FD approximation  $\mathbf{A}$  – while the infinitesimal generator in the KFE  $\mathcal{A}_t^*$  does due to its dependence on  $r_t = r(K_t, Z_t)$  and  $w_t = w(K_t, Z_t)$  – and consequently so will its FD approximation  $\mathbf{A}_t^T$ .

<sup>21</sup>Just as in chapter 4, the matrices used to solve the KFE should be the transpose of the *converged* HJB matrices.

lation to get  $\mathbf{g}_{t+\Delta_t}$  as:

$$\mathbf{g}_{t+\Delta_t} = \omega_{z_t} \omega_{K_t} \mathbf{g}_{t+\Delta_t}^{uu} + (1 - \omega_{z_t}) \omega_{K_t} \mathbf{g}_{t+\Delta_t}^{du} + \omega_{z_t} (1 - \omega_{K_t}) \mathbf{g}_{t+\Delta_t}^{ud} + (1 - \omega_{z_t}) (1 - \omega_{K_t}) \mathbf{g}_{t+\Delta_t}^{dd}.$$

Although at each time step we are now doing four operations, because each submatrix is only of size  $(N_a \cdot N_y) \times (N_a \cdot N_y)$ , each operation will solve extremely fast.

Fourth, in order to avoid dependency of the model solution on initial conditions, we burn the first  $T_0$  periods of the simulation before performing the regression in step 2.d. The value of  $T_0$  should be chosen based both on the model persistence and on how far the initial conditions are from the “ergodic set.” A good choice of the initial value for the distribution  $g_0$  is to use the distribution in either the *deterministic steady state* – the steady state distribution obtained from simulating the equivalent Aiyagari economy – or, even better, the *stochastic steady state* – the steady state distribution obtained from simulation the Krusell-Smith economy with  $z_t = \bar{z} \forall t$ .

Fifth, because there are no theoretical guarantees for convergence on the PLM, implementation of a dampening scheme is usually essential:

$$\Theta^{n+1} = \nu \Theta^n + (1 - \nu) \Phi^n$$

with some dampening parameter  $\nu \in [0, 1)$  usually chosen to be a sufficiently large value. Sometimes adaptive schemes that adjust  $\nu$  based on the convergence history can improve computational performance.

#### 5.2.4. Near-Aggregation

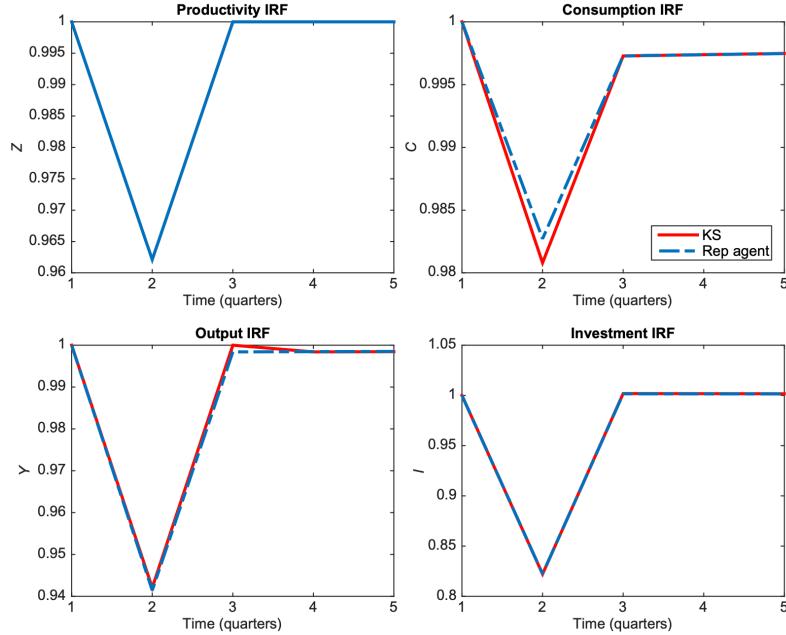
Before examining the implications of near-aggregation in the Krusell and Smith model, it might be useful to define what we actually mean by it: we will say that an economy exhibits **approximate aggregation** if in equilibrium the behavior of aggregate quantities and prices is almost identical to that of a corresponding representative-agent economy. We shall now characterize this approximation more precisely.

In the context of the Krusell and Smith model, we observe that the first moment of the capital distribution,  $K_t$ , effectively serves as a sufficient statistic for forecasting future aggregate quantities and prices. Additional moments of the distribution provide only negligible predictive power. This implies that the heterogeneous-agents economy behaves, to a first-order approximation, as a complete-markets economy would.

To quantify this approximation, consider that the presence of precautionary savings in the Krusell and Smith model increases the aggregate capital stock by merely 0.6% relative to the corresponding representative-agent model. Similarly, the cyclical properties of aggregates exhibit minimal deviation from their representative-agent counterparts, with the consumption-output correlation differing by only 1.5%.

Figure 5.2, taken from Krueger et al. (2016), provides a visual representation of this through impulse response functions (IRFs) to a one standard deviation negative TFP shock in both the RA model and the Krusell-Smith economy. The HA model (solid red lines) and the RA model (dashed blue lines) generate almost identical responses across all

## 5.2. Aggregate Shocks in Continuous Time



**Figure 5.2:** IRF to negative TFP shock – Krueger et al. (2016, Fig. 3)

key macroeconomic variables – consumption, output, and investment.

In response to the approximate aggregation result, one immediate question then arises: if these models generate aggregate dynamics that are essentially equivalent to those of representative-agent models, what justifies the substantial computational complexity they require? The answer, at least for this class of models, is rather limited: while HA models remain essential for addressing distributional questions and understanding welfare implications across different segments of the population, the standard incomplete-markets model provides minimal additional insight regarding aggregate dynamics beyond what representative-agent models already capture.

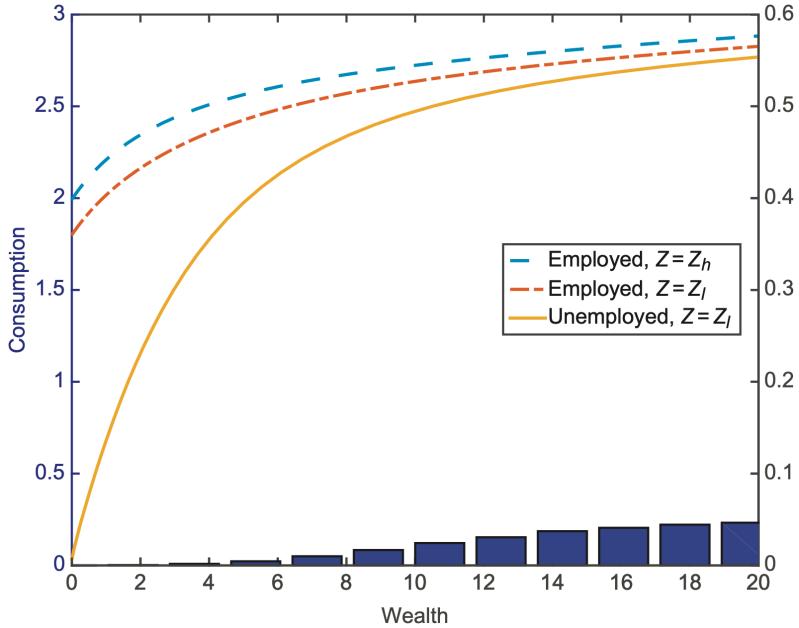
This limitation of what we denoted as “second-generation” models was one of the reasons that motivated the development of subsequent generations of heterogeneous-agent models; models in which – just as in the data – heterogeneity could generate quantitatively meaningful changes in the behavior of aggregates.

## The Mechanism Behind Near-Aggregation

To understand why near-aggregation emerges in the Krusell and Smith model, it is useful to understand what features of the model are responsible of it. The key insight lies in the shape of policy functions and the distribution of wealth across agents.

As shown in section 5.1.2, we know that if the marginal propensity to consume out of wealth is identical across agents, demand aggregation applies. Stated differently, if policy functions are linear

$$a'(a, y; Z, K) = \alpha_0 + \alpha_1 a + \alpha_2 y \quad (5.23)$$



**Figure 5.3:** Consumption and wealth distribution – Krueger et al. (2016, Fig. 5)

tomorrow's aggregate capital can be found as:

$$K' = \int a'(a, y; Z, K) dG = \alpha_0 + \alpha_1 \int adG + \alpha_2 \bar{L} = \tilde{\alpha}_0 + \alpha_1 K \quad (5.24)$$

which yields exact aggregation, with the first moment serving as a sufficient statistic for the distribution  $G$ . However, we know from our analysis in chapter 4 that saving policy functions in HA models exhibit substantial nonlinearity, particularly near borrowing constraints. Why then do we observe approximate aggregation?

The answer to this apparent contradiction lies in the interaction between the nonlinearity of policy functions and the wealth distribution. Figure 5.3, taken from Krueger et al. (2016), illustrates this by overlaying the consumption function with a histogram of the wealth density. We observe can immediately observe that the region of the state space where the marginal propensity to consume (MPC) exhibits the greatest variation – near the borrowing constraint – only contains a negligible fraction of total wealth. Households in this region hold very little assets and thus contribute little to aggregate capital dynamics. On the other hand, the bulk of aggregate wealth is in regions of the state-space where policy functions are approximately linear. These wealthy households, who collectively determine the evolution of aggregate capital, exhibit consumption behavior that closely resembles that of unconstrained agents in a RA framework. The curvature in their policy functions is sufficiently mild that linear approximations provide accurate predictions.

Additionally, because TFP shocks have only limited effects on the wealth distribution, the economy rarely explores regions of the state-space where nonlinearities in households' policy function would generate significant departures from RA dynamics.

**Precautionary Savings and Self-Insurance** — This discussion finally brings us to another question: why are most households in regions of the state-space where policy functions are almost linear? The answer closely relates to the effectiveness of self-insurance in the standard incomplete-markets model. In these models, even access to a single risk-free asset is enough for households to smooth consumption in the face of idiosyncratic income risk. Once households accumulate a modest buffer stock of assets – a process that occurs relatively quickly given the strong precautionary saving motive near the borrowing constraint – their saving behavior becomes predominantly driven by intertemporal substitution considerations rather than precautionary motives. This transition from precautionary to intertemporal saving behavior occurs at wealth levels that are easily attainable for most households who are not perpetually income-constrained. Consequently, the majority of wealth in the economy is held by households whose behavior closely approximates that of unconstrained agents facing no idiosyncratic risk. These households respond to aggregate shocks in a manner that is quantitatively similar to the response of a representative agent, leading to the near-aggregation result.

### Implications for Macroeconomic Modeling

The near-aggregation result raised questions about the appropriate modeling framework for macroeconomic analysis. The failure of second-generation HA models to generate quantitatively significant departures from RA benchmarks can be attributed to specific features of the standard incomplete-markets framework that may not adequately capture empirically relevant sources of heterogeneity.

The available empirical evidence suggests several dimensions along which the standard models fail to match microeconomic data. Most notably, these models generate an average MPC close to the equilibrium interest rate (i.e. approximately 5%), whereas empirical estimates suggest an average MPC closer to 40%, with substantial heterogeneity across households. This discrepancy indicates that the model fails to capture important frictions or constraints that prevent many households from achieving the level of self-insurance implied by the standard framework.

The literature has since explored various extensions designed to generate more empirically plausible distributions of MPCs.<sup>22</sup> These include:

1. Recalibration to target liquid wealth rather than total wealth, recognizing that many assets held by households provide limited consumption-smoothing benefits due to illiquidity constraints.
2. Introduction of ex-ante heterogeneity through preference parameters, such as models with heterogeneous discount factors that generate persistent differences in saving behavior across otherwise similar households.
3. Development of models featuring multiple assets with varying degrees of liquidity, capturing the portfolio constraints that prevent households from fully exploiting intertemporal substitution opportunities.

---

<sup>22</sup>The interested reader can refer to Kaplan and Violante (2022) as a starting point.

4. Incorporation of more realistic income processes and non-linear earnings dynamics that generate greater precautionary saving motives.

Concluding, while the near-aggregation result revealed the limitations of standard incomplete-markets models for understanding aggregate dynamics, it also provided crucial insights into the mechanisms that determine when and how heterogeneity matters for macroeconomic outcomes. The key lesson is that heterogeneity *per se* does not guarantee quantitatively significant departures from RA benchmarks. Rather, the interaction between the distribution of agents across the state space, the shape of policy functions in different regions of that space, and the magnitude of shocks determines whether heterogeneity has first-order implications for aggregate dynamics.

Understanding these mechanisms has proven essential for developing the next generation of HA models attempting to capture both the microeconomic evidence on household behavior and the macroeconomic dynamics observed in the data. The near-aggregation result thus highlighted the need for models that incorporate richer sources of heterogeneity and more realistic frictions to fully capture the feedback between distribution and aggregation.

# Appendices

## A. Deterministic Optimal Control

### A.1. How to draw a phase diagram

Given the system of ODEs as in (1.10) and (1.11) with the associated initial and transversality conditions:

1. Calculate the so called nullclines, by imposing  $\dot{c} = 0$  and  $\dot{k} = 0$
2. Draw the nullcline associated to  $\dot{c} = 0$  in the  $(c, k)$ -space and analyze how consumption moves to each side of this line. In order to do that it is usually sufficient to analyze one of the two cases (e.g.,  $\dot{c} > 0$ ) and to find for which values of  $k$  the inequality holds.
3. Draw the nullcline associated to  $\dot{k} = 0$  in the  $(c, k)$ -space and analyze how capital moves to each side of this line. As before, it is usually enough to analyze one of the two cases and to find for which values of  $c$  the inequality holds.
4. Finally, considering the graph altogether, we will have the direction of every combination of points into the  $(c, k)$ -space, and we will be able to draw all the arrows.

For a more general treatment of phase diagrams, see Hirsch et al. (2004, Chapter 9).

## B. Deterministic HJB Equations

### B.1. The Envelope Theorem

**Theorem 6:** Consider the problem:

$$v(x) = \max_{\alpha} g(x, \alpha)$$

where  $x$  represent the state variable of the problem and  $\alpha$  the control variable. Then:

$$\frac{dv(x)}{dx} = \frac{\partial g(x, \alpha)}{\partial x} \Big|_{\alpha=\alpha^*(x)}$$

with:

$$\alpha^*(x) = \arg \max_{\alpha} g(x, \alpha)$$

*Proof.* Let's note first that:

$$v(x) \equiv g(x, \alpha^*(x)) \quad (25)$$

Taking the total derivative of  $v(x)$ :

$$\frac{dv(x)}{dx} = \frac{\partial g(x, \alpha^*(x))}{\partial x} + \frac{\partial g(x, \alpha^*(x))}{\partial \alpha} \frac{\partial \alpha^*(x)}{\partial x}$$

Note that:

$$\frac{\partial g(x, \alpha^*(x))}{\partial \alpha} = 0$$

by the first-order condition (it is nothing else than the function  $g(x, \alpha)$  evaluated at the optimal control  $\alpha^*(x)$ , and we know that the derivative at that point is 0.) Then:

$$\frac{dv(x)}{dx} = \frac{\partial g(x, \alpha^*(x))}{\partial x} = \frac{\partial g(x, \alpha)}{\partial x} \Big|_{\alpha=\alpha^*(x)}$$

□

In other words what the envelope theorem states is that the derivative of a (value) function with respect to the state variable is equal to the derivative of the function  $g(x, \alpha)$  with respect to the state variable, evaluated at the optimal control  $\alpha^*(x)$ .

In the context of the neoclassical growth model, where we have:

$$\rho v(k) = \max_c u(c) + v'(k) (F(k) - \delta k - c)$$

we can apply the envelope theorem to the value function  $v(k)$ :

$$\rho \frac{dv(k)}{dk} = \frac{\partial g(k, c)}{\partial k} \Big|_{c=c^*(k)}$$

where  $g(k, c) = u(c) + v'(k) (F(k) - \delta k - c)$  and  $c^*(k)$  is the optimal control. Note that

normally we don't write explicitly consumption  $c$  as a function of the state variable  $k$  but it's clear that this is always the case. We get:

$$\rho v'(k) = v''(k) (F(k) - \delta k - c) + v'(k) (F'(k) - \delta)$$

as shown in section 2.1.

## B.2. Non-uniform Grids

In the context of heterogeneous-agent models, it is sometimes useful to discretize the state space on a non-uniform grids. This is because it is often the case that the value function present more curvature closer to the borrowing constraint, while at the same time being close to linear far away from it. Since the FD methods introduced in these notes belong to the class of linear interpolation methods (because they essentially interpolate the value function between grid points using a linear function), the approximation error of these methods increase as the curvature of the function to be approximated increases. Non-uniform grids can therefore be particularly useful in these cases: instead of improving the approximation accuracy by using other (e.g. higher order) methods, which can be slower than linear methods, we instead modify the grid to improve the accuracy only where we need to.

One of the most commonly used non-uniform grids is the **power-spaced grid** where the grid points are spaced according to a power function. We can define the grid as follows:

- ▶ Let  $[\underline{a}, \bar{a}]$  be the possible range for our state variable  $a$
- ▶ Let  $\mathcal{Z}$  be a uniformly spaced grid on  $[0, 1]$ :  $\mathcal{Z} = \{z_i : z_{i+1} = z_i + \Delta \forall i\}$
- ▶ For each grid point  $z_i \in \mathcal{Z}$ , define  $x_i = z_i^\alpha$  for some  $\alpha \in (1, \infty)$  to create a non-uniform grid  $\mathcal{X}$  on  $[0, 1]$ . Note that as  $\alpha \rightarrow \infty$ ,  $\mathcal{X}$  becomes denser and denser closer to 0
- ▶ Then, create asset grid  $\mathcal{A}$  by rescaling each  $x \in \mathcal{X}$

$$a_i = \underline{a} + (\bar{a} - \underline{a})x_i$$

Notice that of course, in the presence of non-uniform grids, our approximation schemes need to be appropriately modified. For example, for the first derivative approximations defined in chapter 2, the scheme becomes:

- ▶ Backward difference:

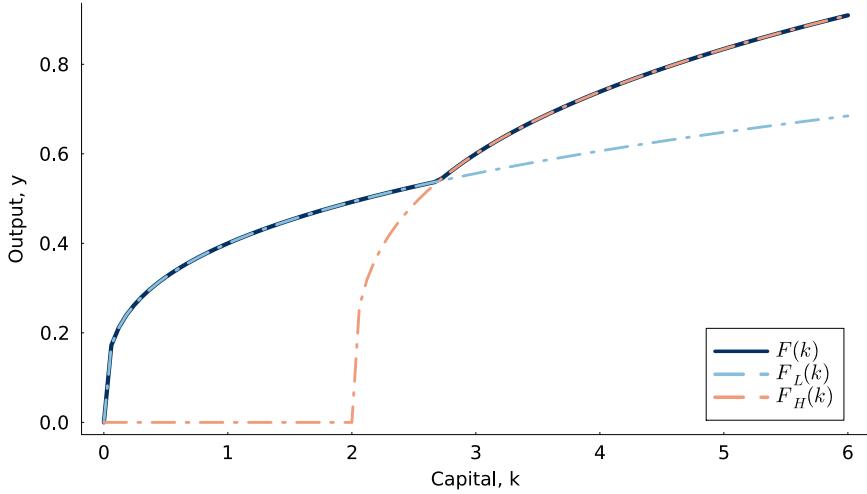
$$v'_{i,B} = \frac{v_i - v_{i-1}}{k_i - k_{i-1}} \approx v'(k_i)$$

- ▶ Forward difference:

$$v'_{i,F} = \frac{v_{i+1} - v_i}{k_{i+1} - k_i} \approx v'(k_i)$$

- ▶ Central difference:

$$v'_{i,C} = \frac{v_{i+1} - v_{i-1}}{k_{i+1} - k_{i-1}} \approx v'(k_i)$$



**Figure 4:** A “butterfly” production function.

### B.3. Non-Convexities

In section 2.2.2 we briefly discussed the need to sometimes perform a Hamiltonian “check” if the value function is (locally or globally) convex. Here we look at a classic example of a model featuring non-convexities in the production function and show how the upwind scheme of section 2.2.2 can be very easily extended to cover such cases.

Let’s consider again the neoclassical growth model:

$$\rho v(k) = \max_c u(c) + v'(k) (F(k) - \delta k - c)$$

but drop the assumption that  $F$  is strictly concave, which we instead define as follows:

$$\begin{aligned} F(k) &= \max \{F_L(k), F_H(k)\} \\ F_L(k) &= A_L k^\alpha \\ F_H(k) &= A_H (\max\{k - \kappa, 0\})^\alpha \end{aligned}$$

where  $\kappa > 0$  and  $A_H > A_L$ . Figure 4 shows how the production function looks like under this assumption.

In discrete time the usual first order condition:

$$u'(F(k) - \delta k - k') = \beta v(k')$$

would no longer be sufficient, because the model typically has multiple solutions. In continuous time, on the other hand, nothing changes and we can continue using the exact same algorithm (and code) that we used to solve the standard version of the model. The only thing we need to add is the Hamiltonian “check” mentioned in section 2.2.2.

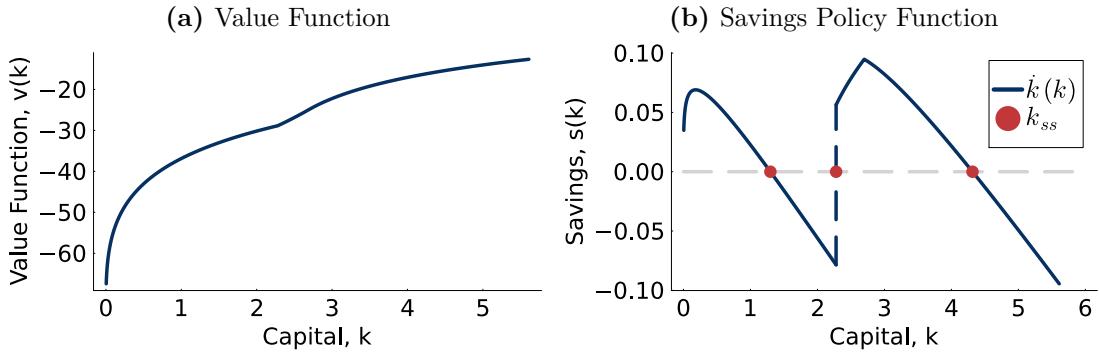
The code snippet in fig. 5 shows how this check is implemented in practice: if the objective is at a maximum – that is, either if  $s_{i,F} \leq 0 \leq s_{i,B}$  or if the steady-state

```

1 # Upwind scheme under non-convexities
2
3 H_F = @. u(c_F) + dv_F*s_F # Forward Hamiltonian
4 H_B = @. u(c_B) + dv_B*s_B # Backward Hamiltonian
5 H_0 = @. u(c_0) # Steady-state Hamiltonian
6
7 # objective is concave
8 I_concave = @. ((s_B < 0) && (s_F <= 0)) || ((s_B >= 0) && (s_F > 0))
9 # objective is convex
10 I_convex = @. ((s_B < 0) && (s_F > 0))
11
12 # When to choose "steady-state"
13 I_max = @. ((s_B >= 0) && (s_F <= 0)) || (H_0 >= max(H_B, H_F))
14 # When to choose backward difference
15 I_B = @. ~I_max && (I_concave && (s_B < 0)) || (I_convex && (H_B >= H_F))
16 # When to choose forward difference
17 I_F = @. ~I_max && (I_concave && (s_F > 0)) || (I_convex && (H_F >= H_B))
18
19 dv_upwind = @. dv_F*I_F + dv_B*I_B + dv_0*I_max
20 c = @. c_F*I_F + c_B*I_B + c_0*I_max
21 # previous line is equivalent to u'_inv.(dv_upwind) but better behaved

```

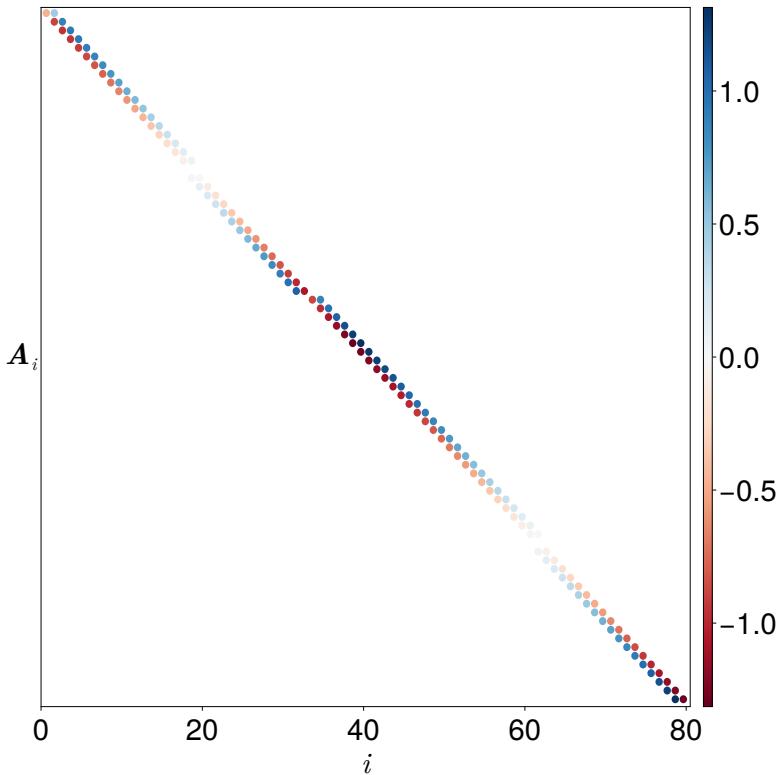
**Figure 5:** Julia code for Hamiltonian adjustment



**Figure 6:** A model with non-convexities

Hamiltonian  $H_{i,0} \equiv u(c_{i,0})$  is larger than both  $H_{i,B}$  and  $H_{i,F}$ , as defined in eqs. (2.8) and (2.9) – the updated upwind scheme uses the “steady-state” solution (i.e. zero-drift); otherwise it chooses the backward difference either if the objective is concave and the backward drift is negative or if the objective is convex and the backward Hamiltonian is greater than the forward Hamiltonian; finally, it chooses the forward difference either if the objective is concave and the forward drift is positive or if the objective is convex and the forward Hamiltonian is greater than the backward Hamiltonian. Once this adjustment is in place, the rest of the code remains exactly identical to the one in section B.4 and the model can be solved using the exact same algorithm.

Figure 6 plots both the value function and the savings policy function of our model. The model has been solved using the same parameters as in section 2.2.5, and the additional parameters  $A_H, A_L$ , and  $\kappa$  have been set to 0.4, 0.6, and 2, respectively. By looking



**Figure 7:** Visualization of the  $\mathbf{A}$  matrix for the model with non-convexities

at panel (a) we can immediately see that the non-convexity in the production function translated also to the value function (which is what required us to adapt the upwind scheme). Somewhat more interestingly – at least from an economic perspective – by analyzing panel (b) we can also characterize the behavior of our economy. As mentioned above, our model will feature multiple equilibria; in particular, the model features two stable equilibria and an unstable one. These steady-states can be found as the points in which the savings policy function (the drift of our state variable,  $\dot{k}$ ) crosses the zero line. For sufficiently high levels of initial capital, the system will monotonically converge to the “good” steady state while, if initial capital is too low, the system will monotonically converge to the “bad” steady state – a classical example of a model featuring a poverty trap. The middle equilibrium is instead unstable because, for any infinitesimally small deviations from the “correct” initial condition, the system will diverge towards either the good or the bad steady state. Finally, to further stress the point that the discretized matrix  $\mathbf{A}$  contains *all* relevant information about the problem, fig. 7 plots the usual visualization for this model – which essentially just shows what we just analyzed looking at the savings policy function.<sup>23</sup>

---

<sup>23</sup>Unlike the rest of the plots in this section, the visualization in fig. 7 has been generated by solving the model on a restricted grid with only 80 points to let all three equilibria feature in the same graph while still being readable.

## B.4. Code for the Neoclassical Growth Model

```

1 ######
2 # Preliminaries
3
4 # You might first need to add packages to your environment if they are not available
5 # You can do so by running: using Pkg; Pkg.add("SparseArrays")
6 using SparseArrays
7
8 #####
9 # Model Parameters
10
11 # Preferences
12 γ = 2          # RRA
13 ρ = 0.05      # Discount rate
14
15 u(c) = (c^(1 - γ) - 1)/(1 - γ)    # Utility function
16 u'(c) = c^(-γ)                      # Marginal utility function
17 u'_inv(x) = x^(-1/γ)                # Inverse of marginal utility function
18
19 # Technology
20 Z = 1          # TFP
21 α = 0.3        # Capital share
22 δ = 0.05      # Depreciation
23
24 # Production function
25 F(k) = Z*k^α
26
27 # Steady state values
28 k_ss = (Z*α/(ρ + δ))^(1/(1 - α))
29 c_ss = F(k_ss) - δ*k_ss
30
31 #####
32 # Grids
33
34 # Capital
35 k_min = k_ss/1000
36 k_max = 2*k_ss
37 Nk = 10_000
38
39 k_grid = range(k_min, k_max, Nk)
40 dk = k_grid[2] - k_grid[1]
41
42 #####
43 # Solver parameters
44
45 max_iter = 1_000
46 tol = 1e-6
47 Δ = 1000
48 iter = 0
49
50 #####
51 # Initialization
52
53 A = SparseMatrixCSC{Float64, Int64}(undef, Nk, Nk)
54
55 v0 = @. u(F(k_grid) - δ*k_grid)/ρ
56 pol_c = similar(v0)
57 pol_s = similar(v0)
58
59 #####

```

```

60 # Model Solution
61
62 while iter <= max_iter
63     iter += 1
64
65     v_diff = diff(v0)/dk
66
67     # Forward difference
68     dv_F = [v_diff; u'(F(k_max) - δ*k_max)]
69     c_F = u'_inv.(dv_F)
70     s_F = @. F(k_grid) - δ*k_grid - c_F
71
72     # Backward difference
73     dv_B = [u'(F(k_min) - δ*k_min); v_diff]
74     c_B = u'_inv.(dv_B)
75     s_B = @. F(k_grid) - δ*k_grid - c_B
76
77     # "Steady-state"
78     c_θ = @. F(k_grid) - δ*k_grid
79     dv_θ = u'.(c_θ)
80
81     # Upwind scheme chooses forward or backward differences based on the sign of the drift
82     I_F = s_F .> 0          # Positive drift -> Forward difference
83     I_B = s_B .< 0          # Negative drift -> Backward difference
84     I_θ = 1 .- I_F .- I_B  # Steady state
85
86     dv_upwind = @. dv_F*I_F + dv_B*I_B + dv_θ*I_θ
87     pol_c = u'_inv.(dv_upwind)
88     pol_s .= @. F(k_grid) - δ*k_grid - pol_c
89     u_vec = u.(pol_c)
90
91     # Elements of the A matrix
92     elem_α = -min.(s_B, 0)/dk    # Lower diagonal
93     elem_ξ = max.(s_F, 0)/dk    # upper diagonal
94     elem_β = - elem_α - elem_ξ # main diagonal
95
96     # Finite difference A matrix
97     A = spdiags([-1 => elem_α[2:end],           # Lower diagonal
98                  0 => elem_β,                   # main diagonal
99                  1 => elem_ξ[1:(end - 1)])  # upper diagonal
100
101    # Check if A matrix is proper
102    maximum(abs.(sum(A, dims = 2))) <= 1e-9 || error("Not all rows of A matrix sum to 0")
103
104    # Implicit scheme
105    B = (ρ + 1/Δ)*SparseArrays.I - A
106    d = u_vec + vθ/Δ
107
108    # Solve system of equations
109    v = B\d
110
111    hjb_diff = maximum(@. abs(v - vθ)/(1 + abs(vθ)))
112    vθ = copy(v)
113
114    println("Iteration #", iter, "; Distance = ", hjb_diff)
115    hjb_diff < tol && break
116 end

```

## C. Stochastic Calculus

### C.1. KFE Derivation

Suppose we have a stochastic process  $X_t \in \mathbb{R}$  that evolves according to the following diffusion:

$$dX = \mu(X, t)dt + \sigma(X, t)dW \quad (26)$$

where  $W_t$  is an independent 1-dimensional Brownian motion process.

We define as  $G_t$  the distribution of state  $X$  at time  $t$ , and as  $g_t$  the corresponding probability density. The goal is to derive a differential equation that defines how the density,  $g_t$ , evolves over time. Such a differential equation is the so-called **Kolmogorov Forward Equation** (KFE) or Fokker-Planck Equation as described in proposition 1:

**Proposition 1** [Kolmogorov Forward Equation]: *Let  $X_t$  be an Itô process described by:*

$$dX = \mu(X, t)dt + \sigma(X, t)dW.$$

*Then,  $g(x, t)$  – the distribution of  $X$  at time  $t$  – satisfies the following Kolmogorov Forward equation:<sup>24</sup>*

$$\frac{\partial g(x, t)}{\partial t} = -\frac{\partial}{\partial x} [\mu(x, t)g(x, t)] + \frac{1}{2} \frac{\partial^2}{\partial x^2} [\sigma^2(x, t)g(x, t)]. \quad (3.12)$$

## Proof

### Step 1: Setup

We start by defining an arbitrary function of our process  $f(x)$ . This function is completely arbitrary and only serves the purpose of being able to apply Itô's lemma. In fact, exactly because it is arbitrary, we will eventually want to get rid of it. We assume that  $f$  is twice continuously differentiable, that it has compact support (i.e. it vanishes outside its support), and that its first and second derivatives also have the same support. That is there exists  $x_1$  and  $x_2$  such that:

$$f(x) = f_x(x) = f_{xx}(x) = 0 \quad \forall x \notin [x_1, x_2].$$

To ease notation, whenever there is no confusion, we will drop the dependencies of functions on  $x$  and  $t$ , so for example  $\mu(x, t)$  and  $\sigma(x, t)$  will simply be written as  $\mu$  and  $\sigma$ . We further assume that the probability density  $g$  is continuous in  $t$  and locally bounded in  $x$ .

---

<sup>24</sup>A derivation of the KFE for univariate diffusion processes can be found in section C.1.

## Step 2: Apply Itô's Lemma and Take Expectations

By applying Itô's lemma we have the usual

$$df = \left( f_x \mu + \frac{1}{2} f_{xx} \sigma^2 \right) dt + f_x \sigma dW.$$

Taking cross-sectional expectations on both sides, i.e., integrating with respect to the density  $g(x, t)$ , we have:

$$\begin{aligned} \mathbb{E}[df] &= \mathbb{E} \left[ \left( f_x \mu + \frac{1}{2} f_{xx} \sigma^2 \right) dt + f_x \sigma dW \right] \\ &= \mathbb{E} \left[ f_x \mu + \frac{1}{2} f_{xx} \sigma^2 \right] dt \end{aligned}$$

or, more explicitly:

$$\begin{aligned} \int_{-\infty}^{\infty} df g(x, t) dx &= \int_{-\infty}^{\infty} \left( f_x \mu + \frac{1}{2} f_{xx} \sigma^2 \right) g(x, t) dt dx + \int_{-\infty}^{\infty} f_x \sigma g(x, t) dW \\ &= \int_{-\infty}^{\infty} \left( f_x \mu + \frac{1}{2} f_{xx} \sigma^2 \right) g(x, t) dt dx \end{aligned}$$

where in the second line we have used the fact that in the cross-section, the expectation of the Brownian motion term is zero.<sup>25</sup>

Separating the right-hand side:

$$\int_{-\infty}^{\infty} df g(x, t) dx = \int_{-\infty}^{\infty} f_x(x) \mu(x, t) g(x, t) dt dx + \frac{1}{2} \int_{-\infty}^{\infty} f_{xx}(x) \sigma^2(x, t) g(x, t) dt dx. \quad (27)$$

## Step 3: Integration by Parts

The “problems” we have now are that on the left-hand side we have the total change in  $f$  and on the right-hand side the derivatives of  $f$ . Because eventually we will want to get rid of  $f$  (which is just an auxiliary function) we start by applying the integration by parts formula, which can be stated as:

$$\int u dv = uv - \int v du.$$

Let's start with the first term on the right-hand side of eq. (27). Set  $dv = f_x(x)dx$  and  $u = \mu(x, t)g(x, t)$ , then integrate  $dv$  to get  $v = f(x)$  and differentiate  $u$  to get  $du = \frac{\partial}{\partial x}(\mu(x, t)g(x, t))dx$ , and plug both into the integration by parts formula:

$$\begin{aligned} \int_{-\infty}^{\infty} f_x(x) \mu(x, t) g(x, t) dt dx &= f(x) \mu(x, t) g(x, t) dt \Big|_{x=-\infty}^{x=\infty} - \int_{-\infty}^{\infty} f(x) \frac{\partial}{\partial x}(\mu(x, t)g(x, t)) dt dx \\ &= - \int_{-\infty}^{\infty} f(x) \frac{\partial}{\partial x}(\mu(x, t)g(x, t)) dt dx \end{aligned}$$

---

<sup>25</sup>Notice that, if  $dW$  were a common noise term, its cross-sectional expectation would not be zero.

where in the second equality we have used the fact that by assumption the function  $f$  vanishes at both ends of the interval and the probability tails down, so the boundary terms vanish.<sup>26</sup>

Now for the second term we will need to apply the same procedure twice. First we set  $dv = f_{xx}(x)dx$  and  $u = \sigma^2(x, t)g(x, t)$ , then we integrate  $dv$  to get  $v = f_x(x)$  and differentiate  $u$  to get  $du = \frac{\partial}{\partial x}(\sigma^2(x, t)g(x, t))dx$ :

$$\begin{aligned}\int_{-\infty}^{\infty} f_{xx}(x)\sigma^2(x, t)g(x, t) dt dx &= f_x(x)\sigma^2(x, t)g(x, t) dt \Big|_{x=-\infty}^{x=\infty} - \int_{-\infty}^{\infty} f_x(x)\frac{\partial}{\partial x}(\sigma^2(x, t)g(x, t)) dt dx \\ &= - \int_{-\infty}^{\infty} f_x(x)\frac{\partial}{\partial x}(\sigma^2(x, t)g(x, t)) dt dx\end{aligned}$$

where again the first term vanishes due to the boundary conditions we imposed on  $f$  and we are left with the second term, which now has been “lowered” by one degree and only includes the first-derivative of  $f$ . We can apply the usual steps once more to get rid of it, which means we eventually have:

$$\int_{-\infty}^{\infty} f_{xx}(x)\sigma^2(x, t)g(x, t) dt dx = \int_{-\infty}^{\infty} f(x)\frac{\partial^2}{\partial x^2}(\sigma^2(x, t)g(x, t)) dt dx.$$

Putting these together, the two terms on the right-hand side of eq. (27) can be written as

$$\int_{-\infty}^{\infty} f(x) \left[ -\frac{\partial}{\partial x}(\mu(x, t)g(x, t)) + \frac{1}{2}\frac{\partial^2}{\partial x^2}(\sigma^2(x, t)g(x, t)) \right] dt dx. \quad (28)$$

Let’s now turn to the left-hand side of eq. (27). By definition of the differential:

$$\int_{-\infty}^{\infty} df g(x, t) dx = \mathbb{E}[df(X_t)] = \mathbb{E} \left[ \lim_{\Delta t \rightarrow 0} (f(X_{t+\Delta t}) - f(X_t)) \right].$$

Since  $f$  is bounded (continuous with compact support), the dominated convergence theorem allows us to exchange the limit and expectation:

$$\begin{aligned}\mathbb{E} \left[ \lim_{\Delta t \rightarrow 0} (f(X_{t+\Delta t}) - f(X_t)) \right] &= \lim_{\Delta t \rightarrow 0} \mathbb{E}[f(X_{t+\Delta t}) - f(X_t)] \\ &= \lim_{\Delta t \rightarrow 0} \left[ \int_{-\infty}^{\infty} f(x)g(x, t + \Delta t) dx - \int_{-\infty}^{\infty} f(x)g(x, t) dx \right].\end{aligned}$$

Combining the integrals and applying the dominated convergence theorem once more

---

<sup>26</sup>Note that even if we do not know the formula for the probability density we can justify this by invoking the fact that the probability that a “particle” travels infinite distance in an infinitesimal amount of time is going to be zero.

(using the facts that  $f$  has compact support and that  $g$  is locally bounded):

$$\begin{aligned}\lim_{\Delta t \rightarrow 0} \left[ \int_{-\infty}^{\infty} f(x) [g(x, t + \Delta t) - g(x, t)] dx \right] &= \int_{-\infty}^{\infty} f(x) \lim_{\Delta t \rightarrow 0} [g(x, t + \Delta t) - g(x, t)] dx \\ &= \int_{-\infty}^{\infty} f(x) dg(x, t) dx\end{aligned}$$

so that the left-hand side of eq. (27) can be simply written as

$$\int_{-\infty}^{\infty} f(x) dg(x, t) dx \tag{29}$$

#### Step 4: Conclude

Equating the left-hand side (29) with the right-hand side (28) we have:

$$\int_{-\infty}^{\infty} f(x) dg(x, t) dx = \int_{-\infty}^{\infty} f(x) \left[ -\frac{\partial}{\partial x} (\mu(x, t)g(x, t)) + \frac{1}{2} \frac{\partial^2}{\partial x^2} (\sigma^2(x, t)g(x, t)) \right] dt dx.$$

Since  $f$  was chosen arbitrarily, for the two integrals to be equal the integrands must be equal, which gives us:

$$dg(x, t) = \left[ -\frac{\partial}{\partial x} (\mu(x, t)g(x, t)) + \frac{1}{2} \frac{\partial^2}{\partial x^2} (\sigma^2(x, t)g(x, t)) \right] dt.$$

Since there is no stochastic term on the right-hand side (the idiosyncratic noise averaged out), the evolution of  $g$  is deterministic. For all values of  $x$ , we therefore have  $dg(x, t) = \frac{\partial g(x, t)}{\partial t} dt$ , and dividing by  $dt$  yields the Kolmogorov Forward Equation:

$$\frac{\partial g(x, t)}{\partial t} + \frac{\partial}{\partial x} (\mu(x, t)g(x, t)) - \frac{1}{2} \frac{\partial^2}{\partial x^2} (\sigma^2(x, t)g(x, t)) = 0.$$

## C.2. $p$ -Variation

A large part of stochastic calculus is concerned about constructing integrals of stochastic processes. In order to do so, however, one needs to have a formal way of articulating how much “variation” a stochastic process has.<sup>27</sup>

**Definition 13** [Partition]: *The set of points  $\mathcal{P} = \{t_0, \dots, t_n\}$  with  $0 = t_0 < t_1 < \dots < t_n = t$  is a partition of the interval  $[0, t]$ . The norm of the partition is  $l(\mathcal{P}) = \max |t_j - t_{j-1}|$ .*

**Definition 14** [ $p$ -Variation]: *Let  $X : \Omega \times \mathcal{T} \rightarrow \mathbb{R}$  be a stochastic process. For any  $p > 0$ , we can define the  $p$ -th variation process of  $X_t$  as the following:*

$$\langle X, X \rangle_t^p(\omega) = \lim_{l(\mathcal{P}) \rightarrow 0} \sum_{j=0}^{n-1} |X_{t_{j+1}}(\omega) - X_{t_j}(\omega)|^p.$$

---

<sup>27</sup>Just as  $X_t$  is shorthand notation for  $X(t)$ ,  $[X]_t$  will be used as shorthand for  $[X](t)$ .

That is,  $\langle X, X \rangle_t^p(\omega)$  is a measure of how much a stochastic process “moves around”, and it is weakly decreasing in  $p$  (i.e. it is easier to have finite  $p$ -variation as  $p$  increases). The intuition is simply that, as  $p$  increases, the term  $|X_{t_{j+1}}(\omega) - X_{t_j}(\omega)|^p$  gets smaller and smaller as the partition gets finer (because  $|X_{t_{j+1}}(\omega) - X_{t_j}(\omega)| \rightarrow 0$  as  $l(\mathcal{P}) \rightarrow 0$ ).

For  $p = 1$ , this is called total variation (or simply variation) process; for  $p = 2$ , this is the **quadratic variation process**. In stochastic calculus quadratic variation and covariation are particularly important. When there is no risk of confusion, we will denote quadratic variation as  $[X]_t$  or  $[X, X]_t$ .

We can also define the quadratic covariation between two stochastic processes  $X, Y$  as:

$$[X, Y]_t = \lim_{l(\mathcal{P}) \rightarrow 0} \sum_{j=0}^{n-1} (X_{t_{j+1}} - X_{t_j})(Y_{t_{j+1}} - Y_{t_j})$$

### Properties of Quadratic Variation<sup>28</sup>

1. Linearity: For stochastic processes  $X, Y, Z$ :

$$[\alpha X + \beta Y, Z]_t = \alpha[X, Z]_t + \beta[Y, Z]_t$$

2. Polarisation identity:

$$[X, Y]_t = \frac{1}{2} ([X + Y, X + Y]_t - [X, X]_t - [Y, Y]_t)$$

3. Jumps of the quadratic covariation process occur only at points where both processes have jumps:

$$\Delta[X, Y]_t = \Delta X_t \Delta Y_t$$

4. If one of the processes  $X$  or  $Y$  is of finite variation, then

$$\Delta[X, Y]_t = \sum_{s \leq t} \Delta X_s \Delta Y_s$$

5. If  $X$  is a continuous stochastic process and  $Y$  is a stochastic process with finite variation, then:

$$[X, Y]_t = 0$$

---

<sup>28</sup>For more details, see Klebaner 2012, section 8.5.

## D. Stochastic HJB Equations

### D.1. Code for the Aiyagari model

Let's start by focusing our attention on the HJB loop in fig. 8, i.e. the part of the code in step 1 of algorithm 4. Relative to the code we used in chapter 2 (showed in its entirety in section B.4) we highlight two main differences: First, we now understand that the boundary conditions imposed in lines 7 and 12 are nothing else than state constraints. In theory, unlike the state constraint for the backward difference in line 12, the boundary condition imposed in line 7 should actually be unnecessary (as long as we solve the problem on a sufficiently large grid) since the upwind scheme should ensure the problem remains in its domain. However, possibly depending on the initial guess or on other problem parameters, it might be the case that *along the convergence path*, one such condition is violated. Hence, it is always a good idea to ensure that reasonable boundary conditions are imposed in the code, even when theoretically unnecessary. In this specific case we impose that the no-savings condition  $\dot{a} = 0$  holds at both boundaries of our state space, which implies  $c = wy + ra$  and  $v_a(a, y) = u'(wy + ra)$ . The second difference is that in lines 33 to 38 we now construct two different generator matrices  $\mathbf{A}_a$ , one for each income state, which will then be added to the main-diagonal blocks in  $\mathbf{A}$  at line 41.

Now, focus on the general-equilibrium loop in fig. 9, i.e. the part of the code in step 2 of algorithm 4 (in line 8 one should actually paste all of the code in fig. 8). In this specific instance we will iterate on the interest rate  $r$  using a bisection algorithm, which is maybe not very sophisticated but is sure to converge. In line 10 we calculate the transpose of  $\mathbf{A}$ , and then solve the eigenvalue problem in lines 13 to 20 by finding the eigenvector associated to the zero-th eigenvalue.<sup>29</sup> Finally, given the stationary distribution we compute the implied capital supply and adjust the interest rate guess accordingly.

---

<sup>29</sup>We here use the same “remove an equation” approach as in section 3.5.1 though any other approach would work just as well

```

1 while iter_hjb <= max_iter_hjb
2     iter_hjb += 1
3     v = copy(v0)
4
5     # Forward difference
6     dv_F[1:(end - 1), :] .= diff(v; dims = 1)/da
7     dv_F[end, :] .= [u'(w*y + r*a_grid[end]) for y in y_grid]
8     c_F = u'_inv.(dv_F)
9     s_F = [w*y + r*a for a in a_grid, y in y_grid] .- c_F
10
11    # Backward difference
12    dv_B[1, :] .= [u'(w*y + r*a_grid[1]) for y in y_grid]
13    dv_B[2:end, :] .= dv_F[1:(end - 1), :]
14    c_B = u'_inv.(dv_B)
15    s_B = [w*y + r*a for a in a_grid, y in y_grid] .- c_B
16
17    # Consumption at steady-state
18    c_0 = [w*y + r*a for a in a_grid, y in y_grid]
19
20    # Upwind scheme based on the sign of the drift
21    I_F = s_F .> 0           # Positive drift -> Forward difference
22    I_B = s_B .< 0           # Negative drift -> Backward difference
23    I_0 = 1 .- I_F .- I_B   # Steady state
24
25    _u = @. u(c_F*I_F + c_B*I_B + c_0*I_0)
26
27    # Elements of the A matrix
28    elem_α = -min.(s_B, 0)/da  # Lower diagonal
29    elem_ξ = max.(s_F, 0)/da  # upper diagonal
30    elem_β = - elem_α - elem_ξ # main diagonal
31
32    # Finite-difference A matrix (for each income state y)
33    A_a = Vector{typeof(A)}(undef, Ny)
34    for iy in 1:Ny
35        A_a[iy] = spdiags([-1 => elem_α[2:end, iy],      # Lower diagonal
36                           0 => elem_β[:, iy],          # main diagonal
37                           1 => elem_ξ[1:(end - 1), iy]) # upper diagonal
38    end
39
40    # Stack A matrices for each income state and add income transition matrix
41    A .= blockdiag(A_a...) + A_y
42
43    # Implicit scheme
44    B = (ρ + 1/Δ)*SparseArrays.I - A
45    d = vec(_u + v/Δ)
46
47    # Solve system of equations
48    v_vec = B\d
49
50    v = reshape(v_vec, Na, Ny)
51    diff_hjb = maximum(abs.(v - v0))
52
53    v0 = copy(v)
54
55    if diff_hjb < tol_hjb
56        break
57    end
58 end
59

```

---

Figure 8: HJB loop in Julia.

```

1 while iter_ge <= max_iter_ge
2     iter_ge += 1
3
4     r = (r_min + r_max)/2
5     KD = (α^Z/(r + δ))^(1/(1 - α))*ȳ
6     w = (1 - α)*Z*(KD/ȳ)^α
7
8     # → HJB Loop here ←
9
10    AT = A'
11
12    # Need to fix one value (any value), otherwise matrix is singular
13    b = zeros(Na*Ny)
14    i_fix = 10
15    b[i_fix] = 0.1
16    AT[i_fix, :] .= 0
17    AT[i_fix, i_fix] = 1
18
19    # Solve system of equations
20    g_vec = AT\b
21    g_ss = reshape(g_vec, Na, Ny)./sum(g_vec.*da) # Renormalize to sum to 1
22
23    # Marginal wrt y
24    g_y = sum(g_ss, dims = 2)
25
26    # Capital supply
27    KS = sum(a_grid.*g_y.*da)
28
29    # Excess supply
30    diff_ge = KS - KD
31
32    # Bisection on the interest rate
33    if diff_ge > tol_ge # Excess supply (i.e. r guess was too high)
34        r_max = r
35    elseif diff_ge < -tol_ge # Excess demand (i.e. r guess was too Low)
36        r_min = r
37    elseif abs(diff_ge) < tol_ge # Equilibrium found
38        break
39    end
40 end

```

**Figure 9:** KFE loop in Julia.

# Bibliography

- Acemoglu, D. (2008). *Introduction to Modern Economic Growth*. Princeton university press.
- Achdou, Y., J. Han, J.-M. Lasry, P.-L. Lions, and B. Moll (2022). “Income and Wealth Distribution in Macroeconomics: A Continuous-Time Approach”. In: *The Review of Economic Studies* 89.1, pp. 45–86.
- Ahn, S., G. Kaplan, B. Moll, T. Winberry, and C. Wolf (2017). “When Inequality Matters for Macro and Macro Matters for Inequality”. In: *NBER Macroeconomics Annual 2017* 32, pp. 1–75.
- Aiyagari, S. R. (1994). “Uninsured Idiosyncratic Risk and Aggregate Saving”. In: *The Quarterly Journal of Economics* 109.3, pp. 659–684.
- Barles, G. and P. E. Souganidis (1991). “Convergence of Approximation Schemes for Fully Nonlinear Second Order Equations”. In: *Asymptotic Analysis* 4.3, pp. 271–283.
- Bewley, T. (1977). “The Permanent Income Hypothesis: A Theoretical Formulation”. In: *Journal of Economic Theory* 16.2, pp. 252–292.
- Bilal, A. (2023). “Solving Heterogeneous Agent Models with the Master Equation”. In: *Working Paper*.
- Cardaliaguet, P., F. Delarue, J.-M. Lasry, and P.-L. Lions (2019). *The Master Equation and the Convergence Problem in Mean Field Games*. Princeton: Princeton University Press.
- Carmona, R. and F. Delarue (2018). *Probabilistic Theory of Mean Field Games with Applications II: Mean Field Games with Common Noise and Master Equations*. Vol. 84. Springer.
- Cherrier, B., P. Duarte, and A. Saïdi (2023). “Household Heterogeneity in Macroeconomic Models: A Historical Perspective”. In: *European Economic Review*, p. 104497.
- Constantinides, G. M. (1982). “Intertemporal Asset Pricing with Heterogeneous Consumers and Without Demand Aggregation”. In: *The Journal of Business* 55.2, pp. 253–267.
- Favilukis, J., S. C. Ludvigson, and S. Van Nieuwerburgh (2016). “The Macroeconomic Effects of Housing Wealth, Housing Finance, and Limited Risk Sharing in General Equilibrium”. In: *Journal of Political Economy* 125.1, pp. 140–223.
- Fernández-Villaverde, J., S. Hurtado, and G. Nuño (2023). “Financial Frictions and the Wealth Distribution”. In: *Econometrica* 91.3, pp. 869–901.

- Gorman, W. M. (1961). "ON A CLASS OF PREFERENCE FIELDS". In: *Metroeconomica* 13.2, pp. 53–56.
- Gu, Z., M. Laurière, S. Merkel, and J. Payne (2023). "Deep Learning Solutions to Master Equations for Continuous Time Heterogeneous Agent Macroeconomic Models". In: *Working Paper*.
- Heathcote, J., K. Storesletten, and G. L. Violante (2009). "Quantitative Macroeconomics with Heterogeneous Households". In: *Annual Review of Economics* 1.1, pp. 319–354.
- Hirsch, M. W., S. Smale, and R. L. Devaney (2004). *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Academic Press. 433 pp.
- Huggett, M. (1993). "The Risk-Free Rate in Heterogeneous-Agent Incomplete-Insurance Economies". In: *Journal of Economic Dynamics and Control* 17.5–6, pp. 953–969.
- Imrohoroglu, A. (1989). "Cost of Business Cycles with Indivisibilities and Liquidity Constraints". In: *Journal of Political Economy* 97.6, pp. 1364–1383.
- Jappelli, T. and L. Pistaferri (2017). *The Economics of Consumption: Theory and Evidence*. Oxford University Press.
- Jarrow, R. and P. Protter (2004). "A Short History of Stochastic Integration and Mathematical Finance: The Early Years, 1880-1970". In: *Lecture Notes-Monograph Series* 45, pp. 75–91.
- Kaplan, G., K. Mitman, and G. L. Violante (2020). "The Housing Boom and Bust: Model Meets Evidence". In: *Journal of Political Economy* 128.9, pp. 3285–3345.
- Kaplan, G., B. Moll, and G. L. Violante (2018). "Monetary Policy According to HANK". In: *American Economic Review* 108.3, pp. 697–743.
- Kaplan, G. and G. L. Violante (2022). "The Marginal Propensity to Consume in Heterogeneous Agent Models". In: *Annual Review of Economics* 14.1, pp. 747–775.
- (2018). "Microeconomic Heterogeneity and Macroeconomic Shocks". In: *The Journal of Economic Perspectives* 32.3, pp. 167–194.
- Karatzas, I. and S. E. Shreve (1998). *Methods of Mathematical Finance*. Vol. 39. Springer.
- (1991). *Brownian Motion and Stochastic Calculus*. 2nd ed. Springer.
- Klebaner, F. C. (2012). *Introduction to Stochastic Calculus with Applications*. 3rd Ed. London: Imperial College Press.
- Kloeden, P. E. and E. Platen (1995). *Numerical Solution of Stochastic Differential Equations*. Springer.
- Krueger, D., K. Mitman, and F. Perri (2016). "Chapter 11 - Macroeconomics and Household Heterogeneity". In: *Handbook of Macroeconomics*. Ed. by J. B. Taylor and H. Uhlig. Vol. 2. Elsevier, pp. 843–921.
- Krusell, P. and A. Smith (1998). "Income and Wealth Heterogeneity in the Macroeconomy". In: *Journal of Political Economy* 106.5, pp. 867–896.
- Kydland, F. E. and E. C. Prescott (1982). "Time to Build and Aggregate Fluctuations". In: *Econometrica: Journal of the Econometric Society*, pp. 1345–1370.
- Ljungqvist, L. and T. J. Sargent (2018). *Recursive Macroeconomic Theory*. MIT press.
- Lucas, R. E. (2003). "Macroeconomic Priorities". In: *American Economic Review* 93.1, pp. 1–14.

- Maliar, L. and S. Maliar (2001). "Heterogeneity in Capital and Skills in a Neoclassical Stochastic Growth Model". In: *Journal of Economic Dynamics and Control* 25.9, pp. 1367–1397.
- (2003). "The Representative Consumer in the Neoclassical Growth Model with Idiosyncratic Shocks". In: *Review of Economic Dynamics* 6.2, pp. 362–380.
- Moll, B. (2024). "The Trouble with Rational Expectations in Heterogeneous Agent Models: A Challenge for Macroeconomics". In: *Working Paper*.
- Moll, B. and L. Ryzhik (2025). *Mean Field Games without Rational Expectations*. Pre-published.
- Munk, C. (2015). *Financial Asset Pricing Theory*. Oxford University Press.
- Negishi, T. (1960). "WELFARE ECONOMICS AND EXISTENCE OF AN EQUILIBRIUM FOR A COMPETITIVE ECONOMY". In: *Metroeconomica* 12.2–3, pp. 92–97.
- Ogaki, M. (2003). "Aggregation under Complete Markets". In: *Review of Economic Dynamics* 6.4, pp. 977–986.
- Øksendal, B. (2014). *Stochastic Differential Equations: An Introduction with Applications*. 6th Ed. Springer.
- Reiter, M. (2009). "Solving Heterogeneous-Agent Models by Projection and Perturbation". In: *Journal of Economic Dynamics and Control* 33.3, pp. 649–665.
- Ross, S. M. (2014). *Introduction to Probability Models*. 10th ed. Academic press.
- Stewart, W. J. (2021). *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press.
- Stokey, N. L., R. E. Lucas Jr., and E. C. Prescott (1989). *Recursive Methods in Economic Dynamics*. Harvard University Press.
- Stokey, N. L. (2009). *The Economics of Inaction: Stochastic Control Models with Fixed Costs*. Princeton University Press.
- Tourin, A. (2013). "Chapter 13 - Introduction to Finite Differences Methods". In: *Optimal Stochastic Control, Stochastic Target Problems, and Backward SDE*. Ed. by N. Touzi. New York, NY: Springer New York, pp. 201–212.