Markov Chain Approximation Method

Kuldeep Singh

Washington University in St. Louis

Nov 2022

Numerical Methods : Continuous Time Models

- Why continuous time numerical methods?

- Why continuous time numerical methods?
  1. Easier to handle constraints

Numerical Methods : Continuous Time Models

- Why continuous time numerical methods?
  1. Easier to handle constraints
  2. Faster

Numerical Methods : Continuous Time Models

- Why continuous time numerical methods?
  1. Easier to handle constraints
  2. Faster
  3. Kolmogorov Forward equation is almost free .

Numerical Methods : Continuous Time Models

- Why continuous time numerical methods?
    1. Easier to handle constraints
    2. Faster
    3. Kolmogorov Forward equation is almost free .
- Requires fixed entry cost, less intuitive

Numerical Methods : Continuous Time Models

- Why continuous time numerical methods?
  1. Easier to handle constraints
  2. Faster
  3. Kolmogorov Forward equation is almost free .
- Requires fixed entry cost, less intuitive

- There are many methods to solve continuous time models. For e.g. Markov Chain Approximation, Finite Difference, Finite Element, Finite Volume.

Today's Plan

- Focus on **Markov Chain Approximation Method** .

Today's Plan

- Focus on **Markov Chain Approximation Method** .
- Understand basic theory/idea via simple example.

Today's Plan

- Focus on **Markov Chain Approximation Method** .
- Understand basic theory/idea via simple example.
- Discussion of computational algorithms.
- Computational details for following problems:
  1. Neoclassical Growth Model
  2. Matlab Code Details for 1
  3. Income Fluctuation Problem

Today's Plan

- Focus on **Markov Chain Approximation Method** .
- Understand basic theory/idea via simple example.
- Discussion of computational algorithms.
- Computational details for following problems:
  1. Neoclassical Growth Model
  2. Matlab Code Details for 1
  3. Income Fluctuation Problem

Main reference

Eslami, Phelan's paper on MCA methods

Markov Chain Approximation Method

Main idea:

1. Approximate the state variables process with a discrete time, finite state Markov chain.
2. HJB equation associated with the problem is approximated by discrete time counterpart.

Markov Chain Approximation Method

Main idea:

1. Approximate the state variables process with a discrete time, finite state Markov chain.
2. HJB equation associated with the problem is approximated by discrete time counterpart.

- Method is fast, can deal with wide variety of problems.
- This approach being closer to discrete time, seems more intuitive.

Theory

- Neoclassical growth model

$$\max_{c_t} \mathbb{E}_0 \int_0^\infty e^{-\rho t} \frac{c_t^{1-\gamma}}{1-\gamma} dt$$

$$dk_t = (f(k_t) - \delta k_t - c_t)dt + \sigma dZ_t$$

$$\rho \tilde{V}(k) = \max_c \frac{c^{1-\gamma}}{1-\gamma} + \tilde{V}'(k)(f(k) - \delta k - c) + \tilde{V}''(k)\frac{\sigma^2}{2}$$

Main Idea

- Approximate $\tilde{V}$ with V as follows

$$V(k) = \max_{c} \frac{\Delta_t c^{1-\gamma}}{1-\gamma} + e^{-\rho\Delta_t}\mathbb{E}V(k')$$

Main Idea

- Approximate $\tilde{V}$ with V as follows

$$V(k) = \max_c \frac{\Delta_t c^{1-\gamma}}{1-\gamma} + e^{-\rho\Delta_t}\mathbb{E}V(k')$$

$$\mathbb{E}V(k') = \sum_{k'} \mathbb{P}(k')V(k')$$

Details

Main Idea

- Approximate $\tilde{V}$ with V as follows

$$V(k) = \max_c \frac{\Delta_t c^{1-\gamma}}{1-\gamma} + e^{-\rho\Delta_t}\mathbb{E}V(k')$$

$$\mathbb{E}V(k') = \sum_{k'} \mathbb{P}(k')V(k')$$

▸ Details

- Can be shown that $V \to \tilde{V}$, if we get the *probabilities right*.

Main Idea

- Approximate $\tilde{V}$ with V as follows

$$V(k) = \max_c \frac{\Delta_t c^{1-\gamma}}{1-\gamma} + e^{-\rho\Delta_t}\mathbb{E}V(k')$$

$$\mathbb{E}V(k') = \sum_{k'} \mathbb{P}(k')V(k')$$

▸ Details

- Can be shown that $V \to \tilde{V}$, if we get the *probabilities right.*
- To get the probabilities right, we need to approximate the law of motion of capital by a Markov chain.

Main Idea

- Approximate $\tilde{V}$ with V as follows

$$V(k) = \max_c \frac{\Delta_t c^{1-\gamma}}{1-\gamma} + e^{-\rho\Delta_t}\mathbb{E}V(k')$$

$$\mathbb{E}V(k') = \sum_{k'} \mathbb{P}(k')V(k')$$

Details

- Can be shown that $V \to \tilde{V}$, if we get the *probabilities right*.
- To get the probabilities right, we need to approximate the law of motion of capital by a Markov chain.
- Formally, probabilities need to satisfy **Local consistency requirement**

In particular, we need[1] :

1. $\mathbb{E}_{t,k,c}(k_{t+\Delta_t} - k) = (f(k) - \delta k - c)\Delta_t + o(\Delta_t)$
2. $\mathbb{E}_{t,k,c}(k_{t+\Delta_t} - k)^2 = \sigma^2 \Delta_t + o(\Delta_t)$

[1] assuming Δ_t to be constant

In particular, we need[1] :

1. $\mathbb{E}_{t,k,c}(k_{t+\Delta_t} - k) = (f(k) - \delta k - c)\Delta_t + o(\Delta_t)$
2. $\mathbb{E}_{t,k,c}(k_{t+\Delta_t} - k)^2 = \sigma^2\Delta_t + o(\Delta_t)$

- Markov chain is constructed such that we get the drift and the diffusion right.

[1] assuming Δ_t to be constant

In particular, we need[1] :

1. $\mathbb{E}_{t,k,c}(k_{t+\Delta_t} - k) = (f(k) - \delta k - c)\Delta_t + o(\Delta_t)$
2. $\mathbb{E}_{t,k,c}(k_{t+\Delta_t} - k)^2 = \sigma^2\Delta_t + o(\Delta_t)$

- Markov chain is constructed such that we get the drift and the diffusion right.

Recall

$$dk_t = (f(k_t) - \delta k_t - c_t)dt + \sigma dZ_t$$

We have,

$$\lim_{\Delta_t \to 0} \mathbb{E}\frac{(k_{t+\Delta_t} - k_t \mid k_t = k)}{\Delta_t} = (f(k_t) - \delta k_t - c_t)$$

$$\lim_{\Delta_t \to 0} \mathbb{E}\frac{(k_{t+\Delta_t} - k_t \mid k_t = k)^2}{\Delta_t} = \sigma^2$$

[1] assuming Δ_t to be constant

Markov Chain

- Capital is restricted a finite set.

$$\{k_1, k_2, ..., k_N\}, \quad k_{i+1} - k_i = \Delta_k$$

Markov Chain

- Capital is restricted a finite set.

$$\{k_1, k_2, ..., k_N\}, \quad k_{i+1} - k_i = \Delta_k$$

- Given $k_t = k_i$,
  we define probabilities for $k_{t+\Delta_t}$ *only* in the set
  $\{k_i, k_i + \Delta_k, k_i - \Delta_k\}$.

Markov Chain

- Capital is restricted a finite set.

$$\{k_1, k_2, ..., k_N\}, \quad k_{i+1} - k_i = \Delta_k$$

- Given $k_t = k_i$,
  we define probabilities for $k_{t+\Delta_t}$ *only* in the set $\{k_i, k_i + \Delta_k, k_i - \Delta_k\}$.
- This is the reason we get speed.
  At k_i, we are just concerned with neighbouring points, unlike usual discrete time problems. As we will see later, this gives us closed form expression for *potential* $c_{optimal}$, given a guess for V

1. First consider, interior points on the grid:

$$\mathbb{P}(k+\Delta_k) = \frac{\Delta_t}{\Delta_k^2}\left[\frac{\sigma^2}{2} + \Delta_k \max\left\{(f(k) - \delta k - c), 0\right\}\right]$$

$$\mathbb{P}(k-\Delta_k) = \frac{\Delta_t}{\Delta_k^2}\left[\frac{\sigma^2}{2} + \Delta_k \max\left\{-(f(k) - \delta k - c), 0\right\}\right]$$

$$\mathbb{P}(k) = 1 - \mathbb{P}(k+\Delta_k) - \mathbb{P}(k-\Delta_k)$$

1. First consider, interior points on the grid:

$$\mathbb{P}(k+\Delta_k) = \frac{\Delta_t}{\Delta_k^2}\left[\frac{\sigma^2}{2} + \Delta_k \max\left\{(f(k) - \delta k - c), 0\right\}\right]$$

$$\mathbb{P}(k-\Delta_k) = \frac{\Delta_t}{\Delta_k^2}\left[\frac{\sigma^2}{2} + \Delta_k \max\left\{-(f(k) - \delta k - c), 0\right\}\right]$$

$$\mathbb{P}(k) = 1 - \mathbb{P}(k+\Delta_k) - \mathbb{P}(k-\Delta_k)$$

2. At upper bound and lower bound of k grid :

$\mathbb{P}(k+\Delta_k) = 0, \mathbb{P}(k-\Delta_k) = 0$ respectively.

Random variable:	$k_{t+\Delta_t} - k_t$		
Values	0	Δ_k	$-\Delta_k$
Probability	$1 - \mathbb{P}_k^+$-$\mathbb{P}_k^-$	$\mathbb{P}_k^+$	$\mathbb{P}_k^-$

$$\mathbb{E}_{t,k,c}(k_{t+\Delta_t} - k_t) = \mathbb{P}(k)0 + \mathbb{P}(k+\Delta_k)\Delta_k - \mathbb{P}(k-\Delta_k)(\Delta_k)$$

$$= (f(k) - \delta k - c)\Delta_t$$

Random variable:	$(k_{t+\Delta_t} - k_t)^2$	
Values	0	Δ_k^2
Probability	$1 - \mathbb{P}_k^+$-$\mathbb{P}_k^-$	$\mathbb{P}_k^+ + \mathbb{P}_k^-$

$$\mathbb{E}_{t,k,c}(k_{t+\Delta_t} - k_t)^2 = \mathbb{P}(k)0 + \left[\mathbb{P}(k+\Delta_k) + \mathbb{P}(k-\Delta_k)\right]\Delta_k^2$$

$$= \sigma^2\Delta_t + \ldots^2$$

[2] ignoring second order terms

Computational Algorithms

- How do we actually compute the value function, consumption policy?

Computational Algorithms

- How do we actually compute the value function, consumption policy?
- Use following algorithms:
  1. Value function iteration
  2. Policy function iteration
  3. Modified policy function iteration
  4. Generalized modified policy function iteration

Comparison

- Results for Income Fluctuation Problem.

Table: Methods Comparison

Method	**Time (secs)**
VFI	5×60
PFI	0.32
MPFI	36

Comparison

- Results for Income Fluctuation Problem.

Table: Methods Comparison

Method	Time (secs)
VFI	5×60
PFI	0.32
MPFI	36

Processor: Intel Core i7-8550U
error tolerance=10^{-8}
$a_{min} = 0$, $a_{max} = 20$, $da = 0.05$, $z_{min} = 0.5$, $z_{max} = 1.50$, $dz = 0.05$

Comparison

- Results for Income Fluctuation Problem.

Table: Methods Comparison

Method	Time (secs)
VFI	5×60
PFI	0.32
MPFI	36

Processor: Intel Core i7-8550U
error tolerance$=10^{-8}$
$a_{min} = 0$, $a_{max} = 20$, $da = 0.05$, $z_{min} = 0.5$, $z_{max} = 1.50$, $dz = 0.05$

- We will focus on VFI and PFI today.

Example I: Neoclassical Growth Model

Let's say, we have a capital grid k_{grid} of N points.

- Value Function Iteration:

1. Start with initial guess V ($N \times 1$ vector).

2. Compute optimal C.

3. Update V_{new}.

4. If $V_{new} \approx V$, stop. Else, start again with $V = V_{new}$ in step 1.

Example I: Neoclassical Growth Model

Let's say, we have a capital grid k_{grid} of N points.

- Value Function Iteration:

1. Start with initial guess V ($N \times 1$ vector).
2. Compute optimal C.
3. Update V_{new}.
4. If $V_{new} \approx V$, stop. Else, start again with $V = V_{new}$ in step 1.

- Never use value function iteration.
- All speed gains undone by the fact that discount factor $e^{-\rho\Delta_t} \approx 1$

- Policy Function Iteration

1. Start with initial guess C ($N \times 1$ vector).
2. Compute V using HJB and C from previous step.
3. Compute optimal consumption, C_{new} for V in step 2.
4. If $C_{new} \approx C$, stop. Else, start again with $C = C_{new}$ in step 1.

Details

$$\Delta_t U_i + e^{-\rho\Delta_t}\left[\mathbb{P}^{+}_{k_i} V_{i+1} + \mathbb{P}^{-}_{k_i} V_{i-1} - (\mathbb{P}^{+}_{k_i} + \mathbb{P}^{-}_{k_i}) V_i + V_i\right] \quad -\mathrm{I}$$

▸ HJB

Details

$$\Delta_t U_i + e^{-\rho\Delta_t}\left[\mathbb{P}^{+}_{k_i} V_{i+1} + \mathbb{P}^{-}_{k_i} V_{i-1} - (\mathbb{P}^{+}_{k_i} + \mathbb{P}^{-}_{k_i}) V_i + V_i\right] \quad -\mathrm{I}$$

HJB Expanding further

$$\max_c \frac{\Delta_t c_i^{1-\gamma}}{1-\gamma} + e^{-\rho\Delta_t}\frac{\Delta_t}{\Delta_k}\max\left\{f(k_i) - \delta k_i - c_i, 0\right\} V^F$$

$$-e^{-\rho\Delta_t}\frac{\Delta_T}{\Delta_k}\max\left\{-(f(k_i) - \delta k_i - c_i), 0\right\} V^B +$$

$$e^{-\rho\Delta_t}\frac{\Delta_t}{\Delta_k^2}\frac{\sigma^2}{2}\left[V^F - V^B\right] + e^{-\rho\Delta_t} V_i$$

$$V^F = V_{i+1} - V_i \quad V^B = V_i - V_{i-1}$$

- Note at optimal c_i^*, we will have one of the following cases:

$$f(k_i) - \delta k_i > c_i^*, \quad f(k_i) - \delta k_i < c_i^*, \quad c_i^* = f(k_i) - \delta k_i$$

- We guess following candidates as maximizers of RHS of equation on last slide.

$$c^F = \left[\frac{e^{-\rho\Delta_t} V^F}{\Delta_k}\right]^{-1/\gamma}$$

$$c^B = \left[\frac{e^{-\rho\Delta_t} V^B}{\Delta_k}\right]^{-1/\gamma}$$

$$c^S = f(k) - \delta k$$

In computations it's convenient to define following $N \times 1$ vectors:

$$F := \frac{\Delta_t (C^F)^{1-\gamma}}{1-\gamma} + e^{-\rho\Delta_t}\frac{\Delta_t}{\Delta_k}\max\left\{f(k) - \delta k - C^F, 0\right\}V^F$$
$$-e^{-\rho\Delta_t}\frac{\Delta_t}{\Delta_k}\max\left\{-(f(k) - \delta k - C^F), 0\right\}V^B$$

$$B := \frac{\Delta_t (C^B)^{1-\gamma}}{1-\gamma} + e^{-\rho\Delta_t}\frac{\Delta_t}{\Delta_k}\max\left\{f(k) - \delta k - C^B, 0\right\}V^F$$
$$-e^{-\rho\Delta_t}\frac{\Delta_t}{\Delta_k}\max\left\{-(f(k) - \delta k - C^B), 0\right\}V^B$$

$$S := \frac{\Delta_t (C^S)^{1-\gamma}}{1-\gamma}$$

Then $C = C^i$ is the solution, where $i = max\{F, B, S\}$

- Check out extra section to see how to handle boundary points in computation.

- In computations, you can compactly write **I** as

$$\Delta_t U_{N\times 1} + e^{\rho\Delta_t}(I + A)V$$

- In computations, you can compactly write **I** as

$$\Delta_t U_{N\times 1} + e^{\rho\Delta_t}(I + A)V$$

$$A = \begin{bmatrix} \mathbb{M}_1 & \mathbb{P}^+_{k_1} & 0 & \dots & 0 \\ \mathbb{P}^-_{k_2} & \mathbb{M}_2 & \mathbb{P}^+_{k_2} & 0 & \vdots \\ 0 & \mathbb{P}^-_{k_3} & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \mathbb{P}^+_{k_{N-1}} \\ 0 & \dots & \dots & \mathbb{P}^-_{k_N} & \mathbb{M}_N \end{bmatrix}_{N\times N}$$

$$\mathbb{P}^+_K = \begin{bmatrix} \mathbb{P}^+_{k_1} \\ \mathbb{P}^+_{k_2} \\ . \\ . \\ . \\ 0 \end{bmatrix} \quad \mathbb{P}^-_K = \begin{bmatrix} 0 \\ \mathbb{P}^-_{k_2} \\ . \\ . \\ . \\ \mathbb{P}^-_{k_N} \end{bmatrix} \quad \mathbb{M} := -(\mathbb{P}^+_K + \mathbb{P}^-_K)$$

- Any mistake in A matrix $\Rightarrow$ no convergence or convergence to some nonsensical result.

- Any mistake in A matrix $\Rightarrow$ no convergence or convergence to some nonsensical result.
- Notice: A matrix has all elements 0 except **3 diagonals.**

1. Upper diagonal captures movement to $+\Delta_k$
2. Lower diagonal captures movement to $-\Delta_k$
3. Middle diagonal captures movement to the same point.

Example II: Income Fluctuation Problem

$$\max_{c_t} \mathbb{E}_0 \int_0^\infty e^{-\rho t} \frac{c_t^{1-\gamma}}{1-\gamma} dt$$

$$da_t = (ra_t + z_t - c_t)dt$$

$$dz_t = \mu_{z,t} dt + \sigma_{z,t} dW_t$$

$$a_t \geq -\phi$$

- Just like before, we deal with :

$$V(a,z) = \max_{c \geq 0} \frac{\Delta_t c^{1-\gamma}}{1-\gamma} + e^{-\rho \Delta_t} \sum P(a',z') V(a',z')$$

- Approximate process for a_t, z_t with finite state Markov chains.

$$\mathbb{P}(a \pm \Delta_a, z) = \frac{\Delta_t}{\Delta_a} \max\left\{ \pm (ra + z - c), 0 \right\}$$

$$\mathbb{P}(a, z \pm \Delta_z) = \frac{\Delta_t}{\Delta_z^2} \left\{ \frac{\sigma^2}{2} + \Delta_z \max\{\pm \mu, 0\} \right\}$$

$$\mathbb{P}(a, z) = 1 - \mathbb{P}_a^+ - \mathbb{P}_a^- - \mathbb{P}_z^+ - \mathbb{P}_z^-$$

- Verify that above probabilities satisfy local consistency.

Notice we have a constraint $a_t \geq -\phi$.

- No need to worry about it. Start your asset grid at $-\phi$. Boundary probability conditions will ensure the agent never violates this constraint.

Notice we have a constraint $a_t \geq -\phi$.

- No need to worry about it. Start your asset grid at $-\phi$. Boundary probability conditions will ensure the agent never violates this constraint.
- You can again write **I** as

$$\Delta_t U_{N\times 1} + e^{\rho\Delta_t}(I + A)V$$

- Matrices are naturally bigger now. But everything works same as before.

$$V = \left[V_{11}, V_{21}, \ldots, V_{I1}, V_{12}, \ldots, V_{I2}, \ldots, V_{1J}, \ldots, V_{IJ}\right]'$$

$I :=$ total points on assets grid and $J :=$ total points on shocks grid.

- A matrix is a bit more complicated but very sparse.

$$A = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2}^{z+} & 0 & \dots & 0 \\ \mathbf{A}_{2,1}^{z-} & \mathbf{A}_{2,2} & \mathbf{A}_{2,3}^{z+} & 0 & \vdots \\ 0 & \mathbf{A}_{3,2}^{z-} & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \mathbf{A}_{J-1,J}^{z+} \\ 0 & \dots & \dots & \mathbf{A}_{J,J-1}^{z-} & \mathbf{A}_{J,J} \end{bmatrix}_{(I\times J)\times(I\times J)}$$

- Each $\mathbf{A}$ is a $I \times I$ matrix.

$$\mathbf{A}_{1,1} = \begin{bmatrix} \mathbb{M}_{1,1} & \mathbb{P}_{a_1^+,z_1} & 0 & \dots & 0 \\ \mathbb{P}_{a_2^-,z_1} & \mathbb{M}_{2,1} & \mathbb{P}_{a_2^+,z_1} & 0 & \vdots \\ 0 & \mathbb{P}_{a_3^-,z_1} & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \mathbb{P}_{a_{I-1}^+,z_1} \\ 0 & \dots & \dots & \mathbb{P}_{a_{I-1}^-,z_1} & \mathbb{M}_{I,1} \end{bmatrix}_{I\times I}$$

$$
\mathbf{A}^{\mathbf{z}+}_{\mathbf{1},\mathbf{2}} = \begin{bmatrix} \mathbb{P}_{a_1,z_1^+} & 0 & 0 & \dots & 0 \\ 0 & \mathbb{P}_{a_2,z_1^+} & 0 & 0 & \vdots \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & \mathbb{P}_{a_I,z_1^+} \end{bmatrix}_{I\times I}
$$

$$
\mathbf{A}^{\mathbf{z}-}_{\mathbf{2},\mathbf{1}} = \begin{bmatrix} \mathbb{P}_{a_1,z_2^-} & 0 & 0 & \dots & 0 \\ 0 & \mathbb{P}_{a_2,z_2^-} & 0 & 0 & \vdots \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & \mathbb{P}_{a_I,z_2^-} \end{bmatrix}_{I\times I}
$$

Middle diagonal looks like $\mathbb{M} := -[\mathbb{P}_a^+ + \mathbb{P}_a^- + \mathbb{P}_z^+ + \mathbb{P}_z^-]_{(I\times J)\times 1}$

- Notice: A matrix has all elements 0 except **5 diagonals.**

1. Upper diagonals captures movement to $+\Delta_a, +\Delta_z$
2. Lower diagonal captures movement to $-\Delta_a, -\Delta_z$
3. Middle diagonal captures movement to the same point.

Extra

This is based on neoclassical growth model, can be easily generalized to more state variables.

- Notice, only valid choices at k_1 is between F_1, S_1 as $\mathbb{P}^-_{k_1} = 0$
- Similarly, only valid choices at k_N is between B_N, S_N as $\mathbb{P}^+_{k_N} = 0$
- To deal with this issue in computational friendly manner, I do following:
  - I set F term $-\kappa$ at k_N and B term $-\kappa$ at k_1, where κ is a big positive number.
  - This ensures we don't choose C^F at k_N and C^B at k_1.

- Another concern is $V^B(1), V^F(N)$ is not defined.
- We need them for e.g, when we compare F_1, S_1 and B_N, S_N.
- I set $V^B(1) = \kappa, V^F(N) = -\kappa$, again κ big and positive.

To understand why this makes sense, let's say we want to compare F_1, S_1, B_1. B_1 is ruled out immediately from first bullet point. If choice is $F_1 \Rightarrow \mathbb{P}^+_{k_1} > 0$, then V^B term is 0, so V^B value does not matter. If choice is S_1, $V^B(1)$ value ensures we don't pick up F_1 (write it down to check this).

Common Errors

- Most common error while coding is probabilities > 1 at some grid points.
- This can be avoided by playing with Δ_t a bit. Make it small enough so that probabilities are < 1.

Common Errors

- Most common error while coding is probabilities > 1 at some grid points.
- This can be avoided by playing with Δ_t a bit. Make it small enough so that probabilities are < 1.

- Sometimes initial guess creates a problem too.
- This is generally an issue as we increase state variables.
- To avoid this, play with your initial guess. Make sure it is increasing in resources and makes economic sense.

References

- Achdou, Yves et al. *Income and wealth distribution in macroeconomics: A continuous-time approach*. Tech. rep. National Bureau of Economic Research, 2017.
- “Applications of Markov chain approximation methods to optimal control problems in economics”. In: *Journal of Economic Dynamics and Control* 143 (2022), p. 104437.
- Kushner, Harold Joseph Kushner et al. *Numerical methods for stochastic control problems in continuous time*. Vol. 24. Springer Science & Business Media, 2001.
- Munk, Claus. “The Markov chain approximation approach for numerical solution of stochastic control problems: experiences from Merton’s problem”. In: *Applied Mathematics and Computation* 1.136 (2003), pp. 47–77.

- Computational Appendix: Altruistically-Motivated Transfers Under Uncertainty, Barczyk et.al 2014