Mogensen, Patrick Kofod

**Doctoral Thesis**

# Essays in dynamic economics

PhD Series, No. 214

**Provided in Cooperation with:**
University of Copenhagen, Department of Economics

*Suggested Citation:* Mogensen, Patrick Kofod (2021) : Essays in dynamic economics, PhD Series, No. 214, University of Copenhagen, Department of Economics, Copenhagen

This Version is available at:
https://hdl.handle.net/10419/240562

UNIVERSITY OF COPENHAGEN
FACULTY OF SOCIAL SCIENCES
DEPARTMENT OF ECONOMICS

# Essays in Dynamic Economics

## PhD Thesis

Patrick Kofod Mogensen

Supervisors: Bertel Schjerning

Submitted on: 30th November 2020

# Contents

# Acknowledgements

There are a few distinct moments in my life that caused me to pursue a Ph.D. and they exist as vivid memories in my mind. When I was thirteen years old I read Sophie's World by Jostein Gaarder. I realized that you can use your own mind to create, discuss, improve, and reject ideas and that I wanted to go to university. I was not driven by a subject or field just a fascination with the collectively accumulated stock of ideas and knowledge. A few years later, I was at the Hammershus Castle Ruins on Bornholm with my class and my mathematics and science teacher, Steen Holst. Steen was an excellent teacher and we discussed the different parts of the educational system including university and graduate studies. I realized that I had to give it a go and try to move the frontier of some area of knowledge. Failure would not be the end of the world, but I had to give it a try.

High school came and went, and I enrolled into the economics program at University of Copenhagen and got the grades I needed for graduate school admission. But economics was not facinating to me. I understood it, and pass the exams, but it didn't really interest me. Luckily, I took Adv. Microeconometrics and met Bertel Schjerning. The combination of more advanced theory and computationally oriented classes appealed to me. I remember asking many questions on my way up the stairs in building 26 following Bertel back to his office after classes on my way to the top floor to work as a research assistant for Prof. Christian Groth and Prof. Henrik Hansen. Three mentors: a structural microeconomist, a theoretical macroeconomist, and a development economist with a time series background. All were encouraging and happy to discuss, but I was clearly still without direction. Bertel threw me a bone in the shape of Harold Zurcher - the super-human superintendent able to solve optimal control problems amidst greasy engines and odometers - and I was hooked.

Besides those mentioned by name above, I would be shameful not to thank my wife and three children for guiding me and restoring balance to the force and making me realize that a challenging bug or failing convergence is not a reason to lose sleep or lose sight of the things that matter when all is said and done. I am thankful for my time in Toronto with Prof. Victor Aguirregabiria and for working with all my co-authors. Dennis Kristensen for discussions about the tiniest details of randomized Bellman operators. Bjørn Bjørnsson Meyer for his determination that our blood, sweat, and tears would be worth it in the end. Without Bjørn, I would have likely not handed in a thesis at all.

Patrick Kofod Mogensen

Hvidovre, November 2020

# English Summary

This dissertation contains three self-contained chapters. They are united by the dynamic economic models that are either studied theoretically or applied empirically. The first two were written in collaboration with co-authors.

The *first* chapter is about the theoretical properties of the value function when solving discrete time, discrete choice dynamic programming problems using sieves to approximate the value function. The *second* chapter is about the incentives and dynamics that governs students' progression and work choices. Using a dynamic structural model we explore behavior of university students in Denmark and look into why students generally do not finish on time. We use the model to evaluate a number of counter factual policies affecting university students. The *third* chapter derives equilibrium conditions for directional dynamic games and shows how to solve them using homotopy continuation methods for systems of multivariate polynomials in the complete information formulation of the games and interval arithmetic for the incomplete information games.

**Chapter 1 – Solving Dynamic Discrete Choice Models Using Smoothing and Sieve Methods**

*with Dennis Kristensen, Jong-Myun Moon, and Bertel Schjerning*

*Forthcoming in Journal of Econometrics*[1]

We propose to combine smoothing, simulations and sieve approximations to solve for either the integrated or expected value function in a general class of dynamic discrete choice (DDC) models. We use importance sampling to approximate the Bellman operators defining the two functions. The random Bellman operators, and therefore also the corresponding solutions, are generally non-smooth which is undesirable. To circumvent this issue, we introduce smoothed versions of the random Bellman operators and solve for the corresponding smoothed value functions using sieve methods. We also show that one can avoid using sieves by generalizing and adapting the "self-approximating" method to our setting. We provide an asymptotic theory for both approximate solution methods and show that they converge with $\sqrt{N}$-rate, where $N$ is number of Monte Carlo draws, towards Gaussian processes. We examine their performance in practice through a set of numerical experiments and find that

[1] Available online now. DOI: https://doi.org/10.1016/j.jeconom.2020.02.007

both methods perform well with the sieve method being particularly attractive in terms of computational speed and accuracy.

**Chapter 2 – Student Choices, Incentives, and Labor Markets Outcomes: The Case of Delayed Graduation**

*with Bjørn Bjørnsson Meyer*

In this chapter, we set up a dynamic choice model describing how various pecuniary and non-pecuniary incentives influence university students' decisions on part-time work, dropout, and delayed graduation. We estimate the model using Danish register micro data combined with administrative data from the country's largest university. Counterfactual simulations using the estimated model show that: (i) About half of the average delay in time-to-graduation can be explained by students following economic incentives to prepare for the labor market with work experience. The other half is due to a range of factors, such as income through part-time work and grants and the cost of effort for heavy course load. (ii) Cutting financial aid with one year reduces average time-to-graduation by 0.3 year, but also increases dropout.

**Chapter 3 – Equilibrium Conditions and Solution Methods for Directional Dynamic Oligopoly Games**

In this paper, I derive equilibrium conditions for sub-stages in directional dynamic games with different model specifications in terms of number of actions, number of players, and exogenous (non-)directional states. I show how to use these to solve for all Markov Perfect Equilibria using Recursive Lexicographical Search. I add to the existing literature by deriving the needed equilibrium conditions needed to solve these games, and provide details on how to solve the sub-stages. I show how to solve the system of multivariate polynomial equations in complete information games using all-solution methods and propose a way to solve the more complex system of equations using interval arithmetic in incomplete information versions of some of the games. Full solution methods are important if the aim is to characterize the potential market configurations that can obtain, or if the goal is to estimate structural parameters in a model of dynamic, strategic interaction.

# Danish Summary

Denne PhD-afhandling består af tre selvstændige kapitler. The forenes af dynamiske økonomiske modeller som enten behandles teoretisk eller i en anvendt sammenhæng. De første to kapitler er skrevet sammen med medforfattere. Det andet kapitel indgik også i min medforfatters afhandling.

Det *første* kapitel handler om de teoretiske egenskaber ved værdifunktionen når man løser dynamiske programmeringsmodeller med diskret tid og diskrete valg når *sieves* bruges til at approksimere værdifunktionen. Det *andet* kapitel handler om incitamenter og de dynamiske overvejelser studerende gør sig når de melder sig til fag og søge studiejob imens de er på universitetet. Ved brug af en dynamisk strukturel model undersøger vi de danske studerendes adfærd og kigger på hvorfor de studerende ikke bliver færdige til normeret tid. Vi bruger modellen til at evaluere et par kontrafaktiske simulationer for at evaluere to forslag til politikændringer. The *tredje* kapitel udleder ligevægtsbetingelser for dynamiske spil hvor tilstandsvariablene har en indlejret retningsbestemmelse. Jeg viser hvordan man løser de forskellige modeller med udførlige trin og foreslår to metoder til at løse de udledte ligevægtsbetingelser.

**Chapter 1 – Solving Dynamic Discrete Choice Models Using Smoothing and Sieve Methods**

*with Dennis Kristensen, Jong-Myun Moon, and Bertel Schjerning*

*Forthcoming in Journal of Econometrics*[2]

I dette kapitel foreslår vi at kombinere udglatning, simulering, og sieveapproksimationer til at løse den integrerede værdifunktion eller de forventede værdifunktioner i en generel klasse af dynamiske modeller med diskrete valg. Vi bruger importance sampling til at lave en tilnærmelse af Bellmanoperatorerne som definerer de to funktioner. De randomiserede Bellmanoperatorer, og de tilhørende løsninger, er generelt ikke-glatte hvilket er uønsket. For at håndtere dette problem introducerer vi udglattede udgaver af operatorerne og løser for de tilsvarende udglattede værdifunktioner ved brug af sievemetoder. Vi viser også hvordan man kan undgå sieveapproksimationerne ved at generalisere den såkaldte "selv-tilnærmende" metode til vores setup. Vi udleder assyptotisk teory for begge approksimationer og viser de konvergerer med rate $\sqrt{N}$, hvor $N$ er antal Monte Carlo-træk, til Gaussiske processer. Vi

[2]Available online now. DOI: https://doi.org/10.1016/j.jeconom.2020.02.007

undersøger metodernes ydelse i et praktisk eksempel ved brug af numeriske eksperimenter, og ser at begge metoder fungerer tilfredsstillende. Sieve-metoden er særligt attraktiv både når det kommer til ydelse og præcision.

**Chapter 2 – Student Choices, Incentives, and Labor Markets Outcomes: The Case of Delayed Graduation**

*with Bjørn Bjørnsson Meyer*

I dette kapitel opsætter vi en dynamisk choice model, der beskriver hvordan økonomiske og ikke-økonomiske incitamenter påvirker studerendes valg af deltidsarbejde, frafald og forsinket færdiggørelse. Vi estimerer modelen på dansk register mikrodata kombineret med administrativt data fra landets største universitet. Simulationer af modellen viser at: (i) Omkring halvdelen af den gennemsnitlige forsinkelse kan forklares af, at studerende følger de økonomiske incitamenter til at forberede sig til arbejdsmarkedet med arbejdserfaring. Den anden halvdel består af en række faktorer såsom indkomst fra erhvervsarbejde og uddannelsesstøtte og ulempen ved at tage mange kurser samtidigt. (ii) En beskæring af et års uddannelsesstøtte sænker den gennemsnitlige færdiggørelsestid med 0,3 år, men resulterer også i flere afbrudte studieforløb.

**Chapter 3 – Equilibrium Conditions and Solution Methods for Directional Dynamic Oligopoly Games**

I dette kapitel udleder jeg ligevægtsbetingelser i dynamiske spil hvor tilstandsvariablene har en indlejret retningsbestemmelse. Jeg kigger på modeller med forskelligmarkedsstruktur, og behandler tilfældene hvor agenterne har mange valg, hvor der er flere spillere, og hvor der er eksogene (ikke)-retningsbestemte tilstande. Jeg viser hvordan man kan finde alle Markovperfekte ligevægte ved til brug i Recursive Lexicographical Search. Jeg bidrager med viden om nye typer af spil og detaljer om hvordan disse løses. I viser hvordan man kan løse de multivariate polynomiumssystemer som fremkommer i spil hvor spillerne har viden om alle tilstande, og foreslår en retning i forhold til løsning af spil hvor agenterne har privat information om en personlig tilstandsvariabel. Det er vigtigt at kunne finde alle løsninger i dynamisk spil hvis disse skal kunne bruges i både teoretisk og empirisk sammenhæng til at belyse komplekse strategiske interaktioner.

# Chapter 1
# Solving Dynamic Discrete Choice Models Using Smoothing and Sieve Methods

# Solving Dynamic Discrete Choice Models Using Smoothing and Sieve Methods*

Dennis Kristensen† Patrick K. Mogensen‡ Jong Myun Moon§
Bertel Schjerning¶

December 1, 2020

**Abstract**

We propose to combine smoothing, simulations and sieve approximations to solve for either the integrated or expected value function in a general class of dynamic discrete choice (DDC) models. We use importance sampling to approximate the Bellman operators defining the two functions. The random Bellman operators, and therefore also the corresponding solutions, are generally non-smooth which is undesirable. To circumvent this issue, we introduce smoothed versions of the random Bellman operators and solve for the corresponding smoothed value functions using sieve methods. We also show that one can avoid using sieves by generalizing and adapting the "self-approximating" method of Rust (1997b) to our setting. We provide an asymptotic theory for both approximate solution methods and show that they converge with $\sqrt{N}$-rate, where $N$ is number of Monte Carlo draws, towards Gaussian processes. We examine their performance in practice through a set of numerical experiments and find that both methods perform well with the sieve method being particularly attractive in terms of computational speed and accuracy.

**Keywords:** Dynamic discrete choice; numerical solution; Monte Carlo; sieves.

*We would like to thank Mike Keane, John Rust, Victor Aguirregabiria, Lars Nesheim, Aureo de Paula and many other people for helpful comments and suggestions. Kristensen gratefully acknowledges financial support from the ERC (through starting grant No 312474 and advanced grant No GEM 740369). Schjerning gratefully acknowledges the financial support from the Independent Research Fund Denmark (grant no. DFF – 4182-00052) and the URBAN research project financed by the Innovation Fund Denmark (IFD).

†Department of Economics, University College London, Gower Street, London, United Kingdom. E-mail: d.kristensen@ucl.ac.uk. Website: https://sites.google.com/site/econkristensen.

‡Department of Economics, University of Copenhagen, ï¿œster Farimagsgade 5, Building 35, DK-1353 Copenhagen K, Denmark. E-mail: Patrick.Kofod.Mogensen@econ.ku.dk. Webpage: http://www.economics.ku.dk/staff/phd_kopi/?pure=en/persons/374766.

§PIMCO

¶Department of Economics, University of Copenhagen, ï¿œster Farimagsgade 5, Building 35, DK-1353 Copenhagen K, Denmark. E-mail: Bertel.Schjerning@econ.ku.dk. Webpage: http://bschjerning.com/.

# 1 Introduction

Discrete Decision Processes (DDPs) are widely used in economics to model forward-looking discrete decisions. For their implementation, researchers are required to solve the model which generally cannot be done in closed form. Instead, a number of methods have been proposed for solving the model numerically; see, e.g., Rust (2008) for an overview. We propose two novel methods for approximating the solutions to a general class of Markovian DDP models in terms of either the so-called integrated or expected value function. These two functions are relevant for estimation of DDP's and for welfare analysis of policy experiments. Our framework allows for both continuous and discrete state variables, non-separable utility functions and unrestricted dynamics. As such, we cover most relevant models used in empirical work. The proposed implementation of model and estimators are found to be computationally very efficient, and at the same time providing precise results with small approximation errors due to the use of simulations and sieve methods.

Our first proposal proceeds in three steps: First, we develop smoothed simulated versions of the Bellman operators that returns the integrated and expected value functions as fixed points. Next, we approximate the unknown value function by a sieve, that is, a parametric function class, thereby turning the problem into a finite-dimensional one. Finally, we solve for the parameters entering the chosen sieve using projection-based methods. When the chosen sieve is linear in the parameters, the approximate solution can be computed using an iterative procedure where each step is on closed form.

As an alternative to the above sieve-based method, we also adapt and generalize the so-called "self-approximating" method proposed in Rust (1997b) to our setting: We design the importance sampler used in the simulated Bellman operators so that the corresponding expected and integrated value functions can be solved for directly without the use of sieves. In comparison with the sieve approach, the self-approximating solution method has the advantage that it will not suffer from any biases due to function approximations. But at the same time, the importance sampler used in its implementation will generally have a larger variance compared to the class of samplers that can be used for the sieve method. This larger variance also translates into a larger simulation bias of the self-approximating solution due to the non-linear nature of the problem. Thus, neither method strictly dominates the other.

Our two procedures, the sieve-based and self-approximating one, differ from existing proposals in three important aspects: First, we solve for either the integrated or expected value function instead of the value function itself. This reduces the dimensionality of the problem since we integrate out any i.i.d. shocks appearing in the model before solving it. Moreover, while the value function is non-differentiable, the integrated and expected value functions are generally smooth which means that our sieve method performs better compared to existing ones that aim at approximating the value function. Second, we allow for a general class of importance samplers in the simulation of the Bellman operator; these can be designed to reduce variances and biases due to simulations. Third, we smooth the simulated Bellman operator

by replacing the max-function appearing in its expression by a smoothed version where the degree of smoothing is controlled by a parameter akin to the bandwidth in kernel smoothing methods. This is similar to the logit-smoothed accept-reject simulator of probit models as proposed by McFadden (1989); see also Fermanian and Salanie (2004), Kristensen and Shin (2012) and Iskhakov et al. (2017). The smoothing turns the problem of solving for the integrated and expected value functions into differentiable ones. In particular, the exact solutions to the smoothed simulated Bellman equations become smooth as functions of state variables and any underlying structural parameters. This in turn means that standard sieves, such as polynomials, will approximate the exact solutions well and that we can control the error rate due to function approximation. Moreover, if used in estimation, standard numerical solvers can be employed in computing estimators of the structural parameters. The smoothing entails an additional bias but this can be controlled for by suitable choice of aforementioned smoothing parameter.

The smoothing device also facilitates the theoretical analysis of the approximate value functions since it allows us to use a functional Taylor expansion of it. This expansion is then used to analyze the leading numerical error terms of the approximate value functions due to simulations, smoothing and function approximations. In particular, under regularity conditions, we show that the approximate value function will converge weakly towards a Gaussian process which is the first result of its kind to our knowledge. These results allow researchers to, for example, build confidence intervals around the approximate value function and should be useful when analyzing the impact of value function approximation when used in welfare analysis and estimation of structural parameters. They may also be potentially helpful in designing selection rules for number of basis functions and the smoothing parameter.

A numerical study investigates the performance of the solution methods in practice. We implement the proposed methods for the engine replacement model of Rust (1987) and investigate how smoothing, number of basis functions and number of simulations affect the approximation errors. We also investigate how the procedures are affected by the dimensionality of the problem and how derivative-based solvers affect computation times. We find that the sieve method generally performs best of the two methods: It is computationally faster and in most situations provides a better approximation in terms of bias and variance. Moreover, the sieve method is found to also work well in higher dimensions with its bias and variance being fairly stable as we increase the the number of state variables of the model. In contrast, variances of the self-approximating method increase dramatically as the number of state variables increases and so appears to be less robust. Finally, the errors due to simulations and function approximation behave according to theory and are found to vanish at the expected rates.

Our proposed methods share similarities with the ones developed in, amongst others, Arcidiacono et al. (2013), Keane and Wolpin (1994), Munos and Szepesvari (2008), Norets (2012), Pal and Stachurski (2013) and Rust (1997b) who also use simulations and/or sieve methods to solve DDP's. However, except for Keane and Wolpin (1994), the methods proposed in these papers approximate the value function while ours target the integrated or expected value function

which are more well-behaved (smooth) objects and therefore easier to approximate. Moreover, in contrast to the cited papers, we employ importance sampling and smoothing in our implementation which comes with the aforementioned computational advantages. From a theory perspective, we provide a more complete asymptotic analysis of the approximate integrated and expected value functions. On the other hand, Munos and Szepesvari (2008) and Rust (1997b) provide an analysis of the computational complexity of solving for the value function and so the theories of this paper and these studies complement each other.

The remains of the paper are organized as follows: Section 2 introduces a general class of DDP's and their corresponding value functions. In Section 3, we develop our smoothed simulated versions of the Bellman operators that the integrated and expected value functions are fixed points to. We then show how to (approximately) solve these simulated Bellman equations in Section 4. An asymptotic theory of the approximate value function is presented in Section 5, while the results of the numerical experiments are found in Section 6. Appendix A contains some general results for approximate solutions to fixed point problems, while proofs of the main results can be found in Appendix B.

# 2 Model

We consider the following DDP where a single agent at time $t \geq 1$ solves

$$d_t = \arg\max_{d \in \mathcal{D}} \left\{ u(S_t, d) + \beta E \left[ \nu(S_{t+1}) | S_t, d_t = d \right] \right\}, \tag{2.1}$$

where $\mathcal{D} = \{1, ..., D\}$ is the set of alternatives, $u(S_t, d)$ is the per-period utility, $0 < \beta < 1$ is the discount factor, $S_t$ is a set of state variables that follows a controlled Markov process with transition kernel $F_S\left(S_t | S_{t-1}, d_{t-1}\right)$ and the so-called value function $\nu$ solves the following fixed-point problem,

$$\nu(S_t) = \max_{d \in \mathcal{D}} \left\{ u(S_t, d) + \beta E \left[ \nu(S_{t+1}) | S_t, d_t = d \right] \right\}. \tag{2.2}$$

Following Rust (1987) and many subsequent empirical specifications, we assume that $S_t = (Z_t, \varepsilon_t) \in \mathcal{Z} \times \mathcal{E} \subseteq \mathbb{R}^{d_Z} \times \mathbb{R}^{d_\varepsilon}$ where $Z_t$ and $\varepsilon_t$ satisfy the following conditional independence condition,

$$F_S\left(Z_t, \varepsilon_t | Z_{t-1}, \varepsilon_{t-1}, d_{t-1}\right) = F_\varepsilon\left(\varepsilon_t | Z_t\right) F_Z\left(Z_t | Z_{t-1}, d_{t-1}\right).$$

In many cases $F_\varepsilon\left(\varepsilon_t | Z_t\right) = F_\varepsilon\left(\varepsilon_t\right)$ in which case $\varepsilon_t$ is an i.i.d. sequence and so can be thought of as idiosyncratic shocks to utility. If no shocks are present in the model, we can always choose $\varepsilon_t = \emptyset$ to be an empty variable so that $S_t = Z_t$. Throughout, we will assume that the support $\mathcal{Z}$ is a compact set. This is done to simplify the theoretical analysis since it, for example, implies that value functions defined below will lie in the space of bounded functions on $\mathcal{Z}$, $\mathbb{B}\left(\mathcal{Z}\right)$, equipped with the sup-norm, $\|v\|_\infty = \sup_{z \in \mathcal{Z}} |v\left(z\right)|$. At the same time, we allow the support of the error term, $\mathcal{E}$, be unbounded and for both countable and continuously distributed state

variables.

In the above formulation, the model is characterized by the value function $\nu(s)$. However, it is possible to rewrite the models in terms of either the so-called *integrated value function* or the *expected value function* and solve for these instead. These are defined as

$$v(Z_t) = E\left[\nu(Z_t, \varepsilon_t)|Z_t\right] = \int_{\mathcal{E}} \nu\left(Z_t, e\right) dF_\varepsilon\left(e|Z_t\right),$$

and

$$V(Z_t, d_t) = E\left[\nu(Z_{t+1}, \varepsilon_{t+1})|Z_t, \varepsilon_t, d_t\right] = E[v(Z_{t+1})|Z_t, d_t] = \int_{\mathcal{E}} v(z') dF_Z\left(z'|Z_t, d_t\right),$$

respectively, where we have used the conditional independence assumption. Observe that given $v\left(z\right)$, we can recover $V\left(z, d\right) = E[v(Z_{t+1})|Z_t, d_t]$ which in turn can be used to compute $\nu(S_t) = \max_{d\in\mathcal{D}}\left\{u(Z_t, \varepsilon_t, d) + \beta V\left(Z_t, d\right)\right\}$. Thus, there is no loss in focusing on the integrated and expected value function, except that in the former case we need to compute $E[v(Z_{t+1})|Z_t, d_t]$ numerically to obtain the value function. Furthermore, in many cases the expected value function itself is of interest. For example, the conditional choice probabilities, which are needed for counterfactuals and for estimation, take as input the relative expected value function,

$$P\left(d_t = d|Z_t = z\right) = M_{u,d}(\beta\Delta V(z)|z),\ M_{u,d}(r|z) = \frac{\partial M_u(r|z)}{\partial r\left(d\right)},$$

where $\Delta V(z, d) = V(z, d) - V(z, D)$, $d \in \mathcal{D}$, and$M_u(r|z)$ is a generalized version of the so-called social surplus function defined as, for any $r = (r(1), ..., r(D))$,

$$M_u(r|z) = \int_{\mathcal{E}} \max_{d\in\mathcal{D}}\left\{u(z, e, d) + r(d)\right\} dF_\varepsilon(e|z). \tag{2.3}$$

It is also useful in welfare analysis of policy experiments where we wish to see how a policy change will affect the expected present value of lifetime utility (i.e., the expected value function).

Except for a few special cases, analytical expressions of $v$ and $V$ are not available and so numerical approximations have to be employed. We will here develop numerical methods for solving for either $v$ or $V$ instead of $\nu$ for the following reasons: First, $\nu$ is a function of $s = (z, \varepsilon)$ while $V$ and $v$ are functions of $z$ alone and therefore their approximations are lower-dimensional problems. Second, $\nu$ is non-differentiable due to the max-function in (2.2); in contrast, $v(z)$ and $V(z, d)$ are both smooth functions of $z$ if $F_\varepsilon(e|z)$ and $F_Z\left(z'|z, d\right)$ are. If there is no i.i.d. component in the model, $\varepsilon_t = \emptyset$, then $\nu\left(s\right) = \nu\left(z\right) = v(z)$ and so the integrated value function becomes non-smooth. In contrast, $V(z, d)$ remains smooth even in this case. The functions $v$ and $V$ each solves their own fixed-point problem: Taking conditional expectations on both sides of eq. (2.2), $V$ can be expressed as the solution to

$$V(z, d) = \Gamma(V)(z, d), \tag{2.4}$$

where, with $M_u$ defined in eq. (2.3),

$$\begin{aligned}\Gamma(V)(z,d) &= E\left[\max_{d'\in\mathcal{D}}\{u(Z_{t+1},\varepsilon_{t+1},d')+\beta V(Z_{t+1},d')\}\,|Z_t=z,d_t=d\right]\\ &= \int_{\mathcal{Z}}\int_{\mathcal{E}}\max_{d'\in\mathcal{D}}\{u(z',e,d')+\beta V(z',d')\}\,dF_\varepsilon(e|z')dF_Z(z'|z,d)\\ &= \int_{\mathcal{Z}} M_u(\beta V(z')|z')dF_Z(z'|z,d).\end{aligned}$$

Here and in the following, we let $V(z) = (V(z,1),....,V(z,D))'$ denote the $D\times 1$-vector of expected value function and similar for other objects. With this notation, we can represent the fixed-point problem in vector form, $V(z)=\Gamma(V)(z)$, where

$$\Gamma(V)(z) = \int_{\mathcal{Z}} M_u(\beta V(z')|z')dF_Z(dz'|z). \tag{2.5}$$

Next, to derive the fixed-point problem that $v$ solves, again take conditional expectations on both sides of eq. (2.2) but now only condition on $Z_t$ to obtain

$$v(z) = M_u(\beta V(z)|z). \tag{2.6}$$

Combining this with eq. (2.4),

$$v(z) = M_u\left(\beta\int_{\mathcal{Z}} M_u(\beta V(z')|z')dF_Z(dz'|z)\Big|\, z\right) = \bar{\Gamma}(v)(z). \tag{2.7}$$

where

$$\bar{\Gamma}(v)(z) = M_u\left(\beta\int_{\mathcal{Z}} v(z')\,dF_Z(dz'|z)\Big|\, z\right).$$

Under regularity conditions provided below, $\Gamma$ and $\bar{\Gamma}$ are contraction mappings and so $V$ and $v$ are well-defined and unique. The above two transformations of the original problem into the ones for either the integrated or expected value function are particular cases of the general class of transformations analyzed in Ma and Stachurski (2020).

**Example 1.** Consider the special case where $u(Z_{t+1},\varepsilon_{t+1},d) = \bar{u}(Z_{t+1},d)+\lambda\varepsilon_{t+1}(d)$ for some scale parameter $\lambda>0$ and $F_\varepsilon(e|z)=F_\varepsilon(e)$ in which case

$$M_u(r|z) = \int_{\mathcal{E}}\max_{d\in\mathcal{D}}\{\bar{u}(z,d)+\lambda e(d)+r(d)\}\,dF_\varepsilon(e) = G_\lambda(\bar{u}(z)+r),$$

where $G_\lambda(r) := \int_{\mathcal{E}}\max_{d\in\mathcal{D}}\{\lambda e(d)+r(d)\}\,dF_\varepsilon(e)$. Thus,

$$\bar{\Gamma}(v)(z) = G_\lambda\left(\bar{u}(z)+\beta\int_{\mathcal{Z}} v(z')\,dF_Z(z'|z)\right).$$

If $\varepsilon_t(1),...,\varepsilon_t(D)$ are mutually independent and each component follows a suitably normalized

extreme value distribution then (see, e.g., Rust et al., 2002),

$$G_\lambda(r) = \lambda \log \left[ \sum_{d \in \mathcal{D}} \exp\left( \frac{r(d)}{\lambda} \right) \right]. \tag{2.8}$$

# 3 Simulated Bellman operators

As a first step towards a computationally feasible method for solving for either $v$ or $V$, we here develop simulated versions of their two Bellman operators and then introduce the smoothing device. To allow for added flexibility and precision in the implementation and to cover as special case a modified version of Rust's self-approximating solution method, we employ importance sampling: Let $\Phi_Z(z'|z,d)$ and $\Phi_\varepsilon(e|z)$ be conditional importance sampling distribution functions as chosen by the researcher. These have to be chosen such that $F_Z(\cdot|z,d)$ and $F_\varepsilon(\cdot|z)$ are absolutely continuous w.r.t. $\Phi_Z(\cdot|z,d)$ and $\Phi_\varepsilon(\cdot|z)$, respectively, with Radon-Nikodym derivatives $w_Z(\cdot|z,d) \geq 0$ and $w_\varepsilon(\cdot|z) \geq 0$ so that

$$\frac{dF_Z(z'|z,d)}{d\Phi_Z(z'|z,d)} = w_Z(z'|z,d), \quad \frac{dF_\varepsilon(e|z)}{d\Phi_\varepsilon(e|z)} = w_\varepsilon(e|z). \tag{3.1}$$

We will throughout assume that eq. (3.1) is satisfied. In the leading case $dF_Z = f_Z d\mu_Z$ and $d\Phi_Z = \phi_Z d\mu_Z$ for some measure $\mu_Z$ in which case $w_Z = f_Z/\phi_Z$ and similar for the sampling of $\varepsilon_t$. The above covers the case where $F_Z(z'|z,d)$ is a continuous distribution (in which case $\mu_Z$ is the Lesbesque measure), a discrete distribution (in which case $\mu_Z$ is the counting measure) and the mixed case. With discrete finite support, we could in principle compute the exact Bellman equation and its corresponding solution and so would not need to resort to numerical methods. But if the discrete support is large this may still be computationally very demanding and so even in this case the numerical methods developed below may be computationally attractive, c.f. Arcidiacono et al. (2013).

Given the chosen importance sampler, we can rewrite $\Gamma(V)(z,d)$ as

$$\Gamma(V)(z,d) = \int_{\mathcal{Z}} \int_{\mathcal{E}} \max_{d' \in \mathcal{D}} \{u(s',d') + \beta V(z',d')\} w(s'|z,d) \, d\Phi(s'|z,d)$$

where $s' = (z', e')$ and

$$w(s'|z,d) = w_\varepsilon(e'|z') w_Z(z'|z,d), \; \Phi(s'|z,d) = \Phi_\varepsilon(e'|z')\Phi_Z(z'|z,d).$$

For any given candidate $V$, we can then approximate this integral by Monte Carlo methods: First generate $N \geq 1$ i.i.d. draws, $Z_i(z,d) \sim \Phi_Z(\cdot|z,d)$ and $\varepsilon_i(z,d) \sim \Phi_\varepsilon(\cdot|Z_i(z,d))$, $i = 1, ..., N$, and then compute

$$\Gamma_N(V)(z,d) = \sum_{i=1}^{N} \max_{d' \in \mathcal{D}} \{u(S_i(z,d), d') + \beta V(Z_i(z,d), d')\} w_{N,i}(z,d), \tag{3.2}$$

where $S_i(z,d) = (Z_i(z,d), \varepsilon_i(z,d))$ and

$$w_{N,i}(z,d) = \frac{w(S_i(z,d)|z,d)}{\sum_{i=1}^N w(S_i(z,d)|z,d)}. \tag{3.3}$$

Note here that we normalize the importance weights so that $\sum_{i=1}^N w_{N,i}(z,d) = 1$. This is done to ensure that $\Gamma_N$ is a contraction mapping on $\mathbb{B}(\mathcal{Z})^D$. Similarly, we approximate $\bar{\Gamma}(v)$ by

$$\bar{\Gamma}_N(v)(z) = \sum_{j=1}^N \max_{d' \in \mathcal{D}} \left\{ u(z, \varepsilon_j(z,d'), d') + \beta \sum_{i=1}^N v(Z_i(z,d')) w_{Z,N,i}(z,d') \right\} w_{\varepsilon,N,j}(z,d'), \tag{3.4}$$

where again we normalize the weights to ensure $\bar{\Gamma}_N$ is a contraction on $\mathbb{B}(\mathcal{Z})$,

$$w_{Z,N,i}(z,d) = \frac{w_Z(Z_i(z,d)|z,d)}{\sum_{i=1}^N w_Z(Z_i(z,d)|z,d)}, \; w_{\varepsilon,N,i}(z,d) = \frac{w_\varepsilon(\varepsilon_i(z,d)|z)}{\sum_{i=1}^N w_\varepsilon(\varepsilon_i(z)|z)}.$$

When $\varepsilon_t = \emptyset$, the simulated Bellman operator $\bar{\Gamma}_N$ includes as special cases the ones considered in Rust (1997b) (who chooses $\Phi_Z$ as the uniform distribution on $\mathcal{Z}$) and Pal and Stachurski (2013) (who chooses $\Phi_Z = F_Z$).

**Example 1 (continued).** Suppose we can compute the integral w.r.t. $\varepsilon_t$ analytically. In this case, the following simplified version of the simulated Bellman operator can be employed,

$$\bar{\Gamma}_N(v)(z) = G_\lambda \left( \bar{u}(z) + \beta \sum_{i=1}^N v(Z_i(z)) w_{Z,N,i}(z) \right), \tag{3.5}$$

where $G_\lambda$ was defined in eq. (2.8). Importantly, the max-function has been replaced by its smoothed version $G_\lambda(\cdot)$.

If $F_\varepsilon(e|z)$ and $F_Z(z'|z,d)$ are smooth functions w.r.t. $z$ then $\Gamma(V)(z)$ and $\bar{\Gamma}(v)(z)$ will be smooth functions of $z$ as well. In contrast, the general versions of $\Gamma_N(V)(z,d)$ and $\bar{\Gamma}_N(v)(z)$ are non-smooth due to the presence of the max-function in their definitions which does not get smoothed for finite $N$. This in turn implies that their corresponding fixed points, $V_N(z) = \Gamma_N(V_N)(z)$ and $v_N(z) = \bar{\Gamma}_N(v_N)(z)$, will be non-differentiable w.r.t. the state variables, $z$, and w.r.t. any underlying structural parameters in the model. This is an unattractive feature for two reasons: First, estimation and counterfactuals will be non-smooth problems. Second, the theoretical analysis of $V_N$ and $v_N$ becomes more complicated.

To resolve this issue, we take inspiration from the additive model in Example 1 and propose to smooth the simulated Bellman operators by replacing the "hard" max-function appearing in eqs. (3.2) and (3.4) by its smoothed version $G_\lambda(r)$ defined in eq. (2.8). This yields the following smoothed simulated operators,

$$\Gamma_N(V)(z;\lambda) = \sum_{i=1}^N G_\lambda(u(S_i(z,d)) + \beta V(Z_i(z,d))) w_{N,i}(z), \tag{3.6}$$

$$\bar{\Gamma}_N(v)(z;\lambda) = \sum_{j=1}^{N} G_\lambda \left( u(z, \varepsilon_j(z)) + \beta \sum_{i=1}^{N} v(Z_i(z)) w_{Z,N,i}(z) \right) w_{\varepsilon,N,j}(z), \tag{3.7}$$

where $u(z, \varepsilon_j(z)) = (u(z, \varepsilon_j(z,1), 1), ..., u(z, \varepsilon_j(z,D), D))$ and other vector functions are defined similarly. Setting $\lambda = 0$ in eqs. (3.6)-(3.7), we recover the original non-smooth versions defined in eqs. (3.2)-(3.4). Thus, the smoothed versions are generalized versions of the original ones. The use of $G_\lambda(r)$ in place of $\max_{d\in\mathcal{D}} r(d)$ generates an additional bias in the approximate solutions, but this can be controlled for by suitable choice of $\lambda$. We now interpret $\lambda > 0$ as a smoothing parameter that plays a role similar to that of the bandwidth in kernel regression estimation. Elementary calculations show

$$0 \leq G_\lambda(r) - \max_{d\in\mathcal{D}} r(d) \leq \lambda \log D, \tag{3.8}$$

so that $G_\lambda(r) \to \max_{d\in D} r(d)$, as $\lambda \to 0$, uniformly in $r \in \mathbb{R}^D$. Thus, the smoothing entails a bias of order $O_P(\lambda)$. We discuss the choice of $\lambda$ in practice in the next section.

In some situations, $G_\lambda(r)$ appears in the Bellman operators as an inherent feature of the model specification in which case no smoothing bias will be present. We saw this in Example 1 and it extends to the following class of models: Suppose that $\varepsilon_t = \left(\varepsilon_t^{(1)}, \varepsilon_t^{(2)}\right)$ with $\varepsilon_t^{(1)} = \left(\varepsilon_t^{(1)}(1), ..., \varepsilon_t^{(1)}(D)\right)$ are mutually independent extreme value shocks that enter the per-period utility additively,

$$d_t = \arg\max_{d\in\mathcal{D}} \left\{ \bar{u}(Z_t, \varepsilon_t^{(2)}, d) + \lambda \varepsilon_t^{(1)}(d) + \beta V(Z_t, d) \right\}, \tag{3.9}$$

where as in Example 1 $\lambda > 0$ is scale parameter that determines the impact of $\varepsilon_t^{(1)}(d)$ on the per-period utility. By the same arguments as in Example 1, we find that the expected value function in this case solves $\Gamma_\lambda(V) = V$ where

$$\Gamma_\lambda(V)(z) = \int_{\mathcal{Z}} \int_{\mathcal{E}} G_\lambda(u(z', e')) + \beta V(z')) \, dF_{\varepsilon^{(-1)}}(e'|z) dF_Z(z'|z, d),$$

and $\Gamma_{N,\lambda}$ in eq. (3.6) is clearly an unbiased simulated version of $\Gamma_\lambda$. Similarly, $\bar{\Gamma}_{N,\lambda}$ is an unbiased estimator of $\bar{\Gamma}_\lambda$. To summarize, if the original model of interest contains an additive extreme value term, which is the case in many empirical papers, $G_\lambda$ appears as part of the model and so no smoothing bias will be present in our proposed simulated Bellman operators.

The above shows that the smoothing device corresponds to adding structural shocks to the DDP of interest. In earlier work on solving DDPs, researchers have in some cases done the opposite and removed structural errors in order to facilitate the numerical solution of the model; see Lumsdaine et al. (1992) for one example of this. This was, however, done in the context of discrete state variables with a small number of support points in which case removing continuous structural errors meant that the Bellman operators could be evaluated analytically. Our method is aimed at models where the state variables are either continuous or have a very large discrete support in which case simulations are required in the first place to evaluate the

Bellman operator. Once simulations are introduced, there is little computational gains from removing shocks from the model and instead the introduction of smoothing facilitates solving and analyzing the corresponding solution.

# 4 Approximate value functions

The smoothed simulated Bellman operators $\Gamma_N(V)(z,\lambda)$ and $\bar{\Gamma}_N(v)(z,\lambda)$ in eqs. (3.6)-(3.7) are functionals that, for given function $V$ and $v$, depend on $(z,\lambda)$. We will here and in the following treat them as functionals that take given function $V(z,\lambda)$ and $v(z,\lambda)$, respectively, and map them into functions of $(z,\lambda) \in \mathcal{Z} \times \left[0,\bar{\lambda}\right]$ for some $\bar{\lambda} > 0$. This simplifies the analysis of the impact of smoothing. In particular, under suitable regularity conditions, $\Gamma_N$ and $\bar{\Gamma}_N$ are contraction mappings on $\mathbb{B}\left(\mathcal{Z} \times \left[0,\bar{\lambda}\right]\right)^D$ and $\mathbb{B}\left(\mathcal{Z} \times \left[0,\bar{\lambda}\right]\right)$, respectively, where $\mathbb{B}\left(\mathcal{Z} \times \left[0,\bar{\lambda}\right]\right)$ denotes the space of bounded functions with domain $\mathcal{Z} \times \left[0,\bar{\lambda}\right]$. Thus, they have unique fixed points $V_N(z,\lambda)$ and $v_N(z,\lambda)$ solving

$$V_N(z,\lambda) = \Gamma_N(V_N)(z,\lambda), \; v_N = \bar{\Gamma}_{N,\lambda}(v_N)(z,\lambda). \tag{4.1}$$

In practice, we will only solve for the particular value of $\lambda$ as chosen by us, but for the theory it proves helpful to treat the solutions as mappings defined on $(z,\lambda) \in \mathcal{Z} \times \left(0,\bar{\lambda}\right)$. However, solving these two simulated Bellman equations are not generally feasible since these are infinite-dimensional problems. We here present two ways to reduce the problems to finite-dimensional ones. The first method is a generalized version of the so-called self-approximating method proposed in Rust (1997b) while the second one uses projection-based methods as advocated by Pal and Stachurski (2013).

## 4.1 Self-approximating method

Rust (1997b) proposed to turn the infinite-dimensional problems in eq. (4.1) into a finite-dimensional ones by choosing the importance sampling to be based on marginal, instead of conditional distributions. In our generalized version this corresponds to restricting $\Phi_Z(z'|z,d) = \Phi_Z(z')$ for some marginal distribution $\Phi_Z(\cdot)$ so that the draws $Z_i \sim \Phi_z(\cdot)$ and $\varepsilon_i \sim \Phi_\varepsilon(\cdot|Z_i)$, $i = 1,...,N$ no longer depend on $(z,d)$. In this case, for a given value of $\lambda \in \left[0,\bar{\lambda}\right]$, the fixed-point problems in eq. (4.1) reduce to the following two sets of $N$ nonlinear equations,

$$V_{N,k} = \sum_{i=1}^{N} G_\lambda\left(u(S_i) + \beta V_{N,i}\right) w_{N,i}(Z_k), \tag{4.2}$$

$$v_{N,k} = \sum_{j=1}^{N} G_\lambda\left(u(Z_k,\varepsilon_j) + \beta \sum_{i=1}^{N} v_{N,i} w_{z,i}(Z_k)\right) w_{\varepsilon,N,j}(Z_k), \tag{4.3}$$

for $k = 1,...,N$, that can be solved for w.r.t. $\{V_{N,\lambda,k} : k = 1,...,N\}$ and $\{v_{N,\lambda,k} : k = 1,...,N\}$, respectively. Here, $V_{N,k} = V_N(Z_k,\lambda)$ and $v_{N,k} = v_{N,\lambda}(Z_k,\lambda)$, $k = 1,...,N$. Each of the two sets

of equations have a unique solution due to the contracting property of $\Gamma_{N,\lambda}$ and $\bar{\Gamma}_{N,\lambda}$. Once, for example, eq. (4.2) has been solved, the approximate expected value functions can be evaluated at any other value $z$ by

$$V_N(z,\lambda) = \sum_{i=1}^{N} G_\lambda\left(u(S_i) + \beta V_{N,i}\right) w_{N,i}(z).$$

Note that $V_N(z,\lambda)$ is a smooth function even if $\lambda = 0$ as long as $w_{N,i}(z)$ is smooth and so smoothing is not needed for this property to hold when marginal samplers are employed. However, without smoothing, the set of equations (4.2) become non-smooth w.r.t. the variables $\{V_{N,k} : k = 1, ..., N\}$ and so cannot be solved using derivative-based methods. Thus, the numerical implementation of the self-approximating method still benefits from smoothing.

In addition to smoothing, the above self-approximating method differs from Rust's original proposal in two other ways: First, while Rust (1997b) solved for the value function $\nu(z,\varepsilon)$, we here solve for either $V(z,\lambda)$ or $v(z,\lambda)$ for a fixed value of $\lambda$. As explained in Section 2, $V$ and $v$ convey the same information as $\nu$ and at the same time they are of lower dimension in terms of variables and are more smooth, features which facilitate their numerical approximation. Moreover, our formulation allows for the following generalized version of the simulated Bellman equations for $v_N$,

$$v_{N,k} = \sum_{j=1}^{\tilde{N}} G_\lambda\left(u(Z_k,\varepsilon_j) + \beta \sum_{i=1}^{N} v_{N,i} w_{z,N,i}(Z_k)\right) w_{\varepsilon,N,j}(Z_k), \tag{4.4}$$

where we allow for different number of draws from $\Phi_\varepsilon$ and $\Phi_Z$. In particular, we can choose $\tilde{N}$ as large as we wish (thereby decreasing the variance of the problem) without increasing the number of variables that need to be solved for $(N)$. A similar generalization of the simulated Bellman equations for $V_N$ is possible. Second, we here only require that the state dynamics together with the chosen importance sampler satisfy (3.1); in contrast, Rust (1997b) assumed that $S_t$ was continuously distributed with compact support and chose as importance sampler the uniform distribution with same support. Thus, our version allows for a broader class of models and samplers.

The self-approximating method may not always work well: First, finding a marginal distribution $\Phi_Z(\cdot)$ so that (3.1) holds can be difficult in some models. For example, in many specifications with continuous dynamics, the transition density $f_Z(z'|z,d)$ of $Z_t$ will have singularities, e.g., $\lim_{z' \to z} f_Z(z'|z,d) = +\infty$, in which case $w_Z(z'|z,d) = f_Z(z'|z,d)/\phi_Z(z')$ is not well-defined no matter how we choose $\phi_Z(z')$. And even if (3.1) does hold, the use of marginal samplers instead of conditional ones will generally lead to a larger variance of the solutions since the "marginal" draws $Z_1, ..., Z_N$ do not adapt to the changing shape of $F_Z(\cdot|z,d)$ as a function of $z$. In particular, many of the draws may fall outside of the support of $F_Z(\cdot|z,d)$ and so are "wasted" in which case a large $N$ is required to achieve a reasonable approximation; see Section 6 for an example of this. This issue tends to become more severe in higher dimensions

($d_Z$ is "large") since the volume of the support shrinks, and so the self-approximating method will generally suffer from a built-in curse-of-dimensionality. This curse-of-dimensionality does not appear in the subclass of models that Rust (1997b) focused on where it was assumed that $S_t|S_{t-1}, d_{t-1}$ has support $[0,1]^{\dim(S_t)}$ for all values of $S_{t-1}, d_{t-1}$.

Finally, given that $V_{N,\lambda}$ and $v_{N,\lambda}$ are solutions to non-linear equations, a large variance in the simulated Bellman operator translates into a large bias as is well-known from non-linear GMM estimators. This can be controlled for by choosing $N$ large. But large $N$ means that numerically solving either (4.2) or (4.3) becomes computationally very costly. These issues motivate us to pursue a sieve-based solution strategy.

## 4.2 Sieve-based method

We now return to the general versions of the simulated Bellman operators and so again allow for conditional importance samplers. Let $\bar{\mathcal{V}} \subseteq \mathbb{B}(\mathcal{Z})$ be a suitable function space that $z \mapsto v_N(z;\lambda)$ defined in (4.1) is known to lie in; see below for more details on this. We then choose a finite-dimensional function space (commonly called a sieve in the econometrics literature) $\bar{\mathcal{V}}_K = \{v_K(\cdot;\alpha) : \mathcal{Z} \mapsto \mathbb{R} | \alpha \in \mathcal{A}_K\} \subseteq \bar{\mathcal{V}}$, where $\mathcal{A}_K \subseteq \mathbb{R}^K$ is a parameter set with $K < \infty$, that provides a good approximation to functions in $\bar{\mathcal{V}}$. Similarly, we let $\mathcal{V} \subseteq \mathbb{B}(\mathcal{Z})^D$ be a space of $D$-dimensional vector functions that the solution $z \mapsto V_N(z;\lambda)$ to (4.1) lie in and $\mathcal{V}_K = \left\{V_K(\cdot;\alpha) : \mathcal{Z} \mapsto \mathbb{R}^D | \alpha \in \mathcal{A}_K\right\} \subseteq \mathcal{V}$ be our sieve for this space. Let

$$\bar{\Pi}_K(v) = \arg\min_{v' \in \bar{\mathcal{V}}_K} \|v - v'\|_{\bar{\mathcal{V}}}, \; \Pi_K(V) = \arg\min_{V' \in \mathcal{V}_K} \|V - V'\|_{\mathcal{V}}, \tag{4.5}$$

be the corresponding projections for given (pseudo-) norms $\|\cdot\|_{\bar{\mathcal{V}}}$ and $\|\cdot\|_{\mathcal{V}}$ as chosen by us as well. We then approximate $V_{N,\lambda}$ and $v_{N,\lambda}$ by the solutions to the projected Bellman equations,

$$\hat{v}_{N,\lambda} = \arg\min_{v \in \bar{\mathcal{V}}_K} \left\|v - \bar{\Pi}_K \bar{\Gamma}_{N,\lambda}(v)\right\|_{\bar{\mathcal{V}}}, \; \hat{V}_{N,\lambda} = \arg\min_{V \in \mathcal{V}_K} \|V - \Pi_K \Gamma_{N,\lambda}(V)\|_{\mathcal{V}}. \tag{4.6}$$

These are finite-dimensional problems of size $K$. When $K$ is small relative to $N$, which will generally be the case, the above problems are computationally much more tractable compared to the corresponding self-approximating ones. Note here that these projection-based approximations are different from the least-squares approximations that would solve $\min_{V \in \mathcal{V}_K} \|V - \Gamma_{N,\lambda}(V)\|_{\mathcal{V}}$ and $\min_{v \in \bar{\mathcal{V}}_K} \left\|v - \bar{\Gamma}_{N,\lambda}(v)\right\|_{\bar{\mathcal{V}}}$, respectively. In particular, by suitable choice of the projection operators, $\Pi_K \Gamma_{N,\lambda}$ and $\bar{\Pi}_K \bar{\Gamma}_{N,\lambda}$ will be contraction mappings w.r.t. $\|\cdot\|_\infty$ guaranteeing that $\hat{V}_{N,\lambda}$ and $\hat{v}_{N,\lambda}$ exist and are unique. The following discussion focuses on the integrated value function approximation since it carries over with only minor modifications to the one of the expected value function. We discuss their numerical implementation in further detail in the subsection below.

The projection operator $\bar{\Pi}_K$ can be thought of as a function approximator with the approximation error being $v - \bar{\Pi}_K(v)$ for a given function $v$. Roughly speaking, the projection-based method approximates $v_N$ in (4.1) by $\hat{v}_N = \bar{\Pi}_K(v_N)$ which incurs an additional sieve approxi-

mation error, $v_N - \bar{\Pi}_K(v_N)$. The smoothness of $v_N$ here proves helpful since many well-known sieves are able to provide good approximations of smooth functions using a low-dimensional space ("small" $K$). Due to these features, our proposed projection-based solutions will generally suffer from quite small additional biases relative to the exact simulated solution. This is in contrast to existing projection-based solution methods, such as the one in Pal and Stachurski (2013), that aim at approximating the value function $\nu(s)$ which is non-differentiable.

The smoothness of $v_N$ here help guiding us in choosing the sieve: It allows us to restrict $\bar{\mathcal{V}}$ to a suitable smoothness class and then import existing approximation methods for smooth functions as developed in the literature on numerical methods and nonparametric econometrics. A leading example is the class of linear function approximations where the finite-dimensional function space takes the form of $\bar{\mathcal{V}}_K = \left\{\alpha' B_K(z) : \alpha \in \mathbb{R}^K\right\}$ for a set of basis functions $B_K(z) = (b_1(z), ..., b_K(z))'$. The basis functions can be chosen as, for example, Chebyshev polynomials or B-splines that are able to approximate smooth functions well. However, other non-linear function space are possible such as wavelets, artificial neural networks and shrinkage-type function approximators such as LASSO, where the additional constraints are imposed on $\alpha$; we refer to Chen (2007) for a general overview of different function approximators and constrained sieve estimators. We also allow for flexibility in terms of the chosen norms $\|\cdot\|_{\bar{\mathcal{V}}}$ with a leading example being $\|v\|_{\bar{\mathcal{V}}} = \sum_{i=1}^{M} v^2(z_i)$ for a set of design points $z_1, ..., z_M \in \mathcal{Z}$. Very often the $M \geq 1$ design points will be chosen in conjunction with the sieve.

The above procedure does not suffer from any of the above mentioned issues of the self-approximating method: We can use conditional importance samplers freely which can be designed to control the variance of the simulated Bellman operators; and the dimension of the problem remains $K$ irrespectively of the number of draws $N$. The main drawback is that unique solutions to eqs. (4.6) do not necessarily exist for a given choice of $N$ and $K$. A sufficient condition for this to hold is that $\bar{\Pi}_K$ is a non-expansive operator w.r.t $\|\cdot\|_\infty$, that is, $\left\|\bar{\Pi}_K(v_1) - \bar{\Pi}_K(v_2)\right\|_\infty \leq \|v_1 - v_2\|_\infty$ for any two functions $v_1, v_2 \in \bar{\mathcal{V}}$, since this translates into $\bar{\Pi}_K \Gamma_N$ being a contraction mapping. However, while $\bar{\Pi}_K$ is non-expansive w.r.t. $\|\cdot\|_{\bar{\mathcal{V}}}$ by definition, it is not necessarily non-expansiveness w.r.t $\|\cdot\|_\infty$. Pal and Stachurski (2013) provide some examples of projections that are non-expansive w.r.t. $\|\cdot\|_\infty$, but these are unfortunately computationally expensive to use in general. But $\bar{\Pi}_K$ will generally be close to non-expansive w.r.t $\|\cdot\|_\infty$ asymptotically as $K \to \infty$ for a wide range of sieves and pseudo-norms in the sense that

$$\left\|\bar{\Pi}_K\right\|_{op,\infty} := \sup_{v \in \bar{\mathcal{V}}, \|v\|=1} \left\|\bar{\Pi}_K(v)\right\|_\infty \leq \sup_{v \in \bar{\mathcal{V}}, \|v\|=1} \left\|\bar{\Pi}_K(v) - v\right\|_\infty + 1,$$

where the first term in the last expression will go to zero in great generality as $K \to \infty$ for many popular sieves (see next subsection for details). Given that $\bar{\Gamma}_N$ is a contraction with Lipschitz coefficient $\beta < 1$, this in turn implies that $\bar{\Pi}_K \bar{\Gamma}_N$ will be a contraction mapping for all $K$ large enough. This will be used in our asymptotic analysis of the algorithm. Unfortunately, it is generally not known how large $K$ should be chosen to ensure $\bar{\Pi}_K \bar{\Gamma}_N$ is a contraction. But in our numerical experiments we did not experience any convergence problems.

## 4.3 Numerical implementation of the two methods

We here discuss in more detail the numerical implementation of the self-approximating and projection-based methods. First, the researcher has to choose the importance sampling distributions, the smoothing parameter $\lambda$ and, in the case of the projection-based method, the function approximation method. Second, given these choices, either eq. (4.3) or (4.6) has to be solved for. As before, the discussion here focuses on solving for the integrated value function since most results and arguments for this case carries over with very minor modifications to the expected value function.

### 4.3.1 Importance sampler

The choices of $\Phi_Z$ and $\Phi_\varepsilon$ determine the variance of $\bar{\Gamma}_N$ and should ideally be tailored to minimize it. In the case of projection-based methods, where we can choose $\Phi_Z$ and $\Phi_\varepsilon$ as conditional distributions, we can rely on the already existing theory for efficient importance sampling for how to do so; see Chapter 3 in Robert and Casella (2013) for an introduction. In our numerical experiments, we did not experiment with different choices and throughout set $\Phi_Z = F_Z$ and $\Phi_\varepsilon = F_\varepsilon$.

In the case of the self-approximating method, the choice of $\Phi_Z$ is restricted to the class of marginal distributions. Generally, this entails a large variance of the corresponding simulated Bellman operators. It will hold in great generality that $(Z_t, d_t)$ has a stationary distribution, say, $F_S^*(z,d)$. In this case, a suitable choice would be the marginal of this, $\Phi_Z(z) = F_S^*(z,D)$. However, the stationary distribution depends on the value function and so is rarely available on closed form; so this strategy requires an initial exploration of the model and its solution. Alternatively, one can try to construct a good approximation of the stationary approximation through a mixture Markov model on the form $\Phi_Z(z') = \sum_{d\in\mathcal{D}} \int \omega_d(z) F_Z(z'|z,d)\, d\mu_Z(z)$ for a set of pre-specified mixture weights $\omega_d(z) \geq 0$. In the numerical experiments, we follow Rust (1997b) and choose $\Phi_Z(z)$ as the uniform distribution on $\mathcal{Z}$ which we conjecture is far from optimal in many cases, and so more research in this direction is needed.

### 4.3.2 Smoothing

The use of $G_\lambda(r)$ in place of $\max_{d\in\mathcal{D}} r(d)$ generally generates an additional bias in the corresponding integrated value function of order $O(\lambda)$. At the same time, the variance of $v_N$ is an increasing function of $\lambda$. Thus, ideally we would like to choose $\lambda$ to balance these two effects. A natural criterion would be to minimize the so-called integrated mean-square-error, $\lambda^* = \arg\min_{\lambda\geq 0} E\left[\int_{\mathcal{Z}} \|v_{N,\lambda}(z) - v(z)\|^2 dF_Z(z)\right]$, where $F_Z(z)$ is a suitably chosen distribution such as the stationary one of $Z_t$. Since $v(z)$ is unknown and we cannot evaluate the expectations, $\lambda^*$ cannot be solved for but cross-validation methods can be used instead. This could in principle be done along the same lines as bandwidth selection for smoothed empirical cdfs, see Bowman et al. (1998). However, this is computationally somewhat burdensome. Moreover, in our numerical experiments we found that the quality of the approximate value

function was quite insensitive to the choice of $\lambda$ and so in practice we recommend using very little smoothing such as $\lambda = 0.01$.

### 4.3.3 Function approximation

As mentioned earlier, many approximation architectures are available in the literature. In our numerical experiments we focus on the class of linear function approximators where $\bar{\mathcal{V}}_K = \left\{\alpha' B_K(z) : \alpha \in \mathbb{R}^K\right\}$ for a set of pre-specified basis functions $B_K(z) \in \mathbb{R}^K$. For a given set of $M \geq 1$ design points in $\mathcal{Z}$, $z_1, ..., z_M$, eq. (4.5) then becomes

$$\bar{\Pi}_K(v)(z) = B_K(z)' \left[\sum_{i=1}^{M} B_K(z_i) B_K(z_i)'\right]^{-1} \sum_{i=1}^{M} B_K(z_i) v(z_i). \tag{4.7}$$

The design points may either be random or deterministic and can be chosen relative to the basis functions to ensure that $\bar{\Pi}_K$ is easy to compute and provides a good approximation for a broad class of functions. The performance of most function approximations will depend on the smoothness of the function of interest.

A standard smooth function class often considered in approximation theory is the following: For any vector $a = (a_1, ..., a_{d_Z}) \in \mathbb{N}_0^{d_X}$, let $D^a f(x) = \partial^{|a|} f(x) / \left(\partial x_1^{a_1} \cdots \partial x_{d_Z}^{a_{d_Z}}\right)$, where $|a| = a_1 + \cdots + a_{d_z}$, be the corresponding partial derivative. For $\alpha > 0$, let $\underline{\alpha} \geq 0$ be the greatest integer smaller than $\alpha$. For any $\underline{\alpha}$ times differentiable function $f(x)$, we then define

$$\|f\|_{\alpha,\infty} = \max_{|a| \leq \underline{\alpha}} \|D^a f\|_\infty + \max_{|a| = \underline{\alpha}} \sup_{x_1 \neq x_2} \frac{|D^a f(x_1) - D^a f(x_2)|}{\|x_1 - x_2\|^{\alpha - \underline{\alpha}}}, \tag{4.8}$$

and let $\mathbb{C}_r^\alpha(\mathcal{X})$ be the space of all $\underline{\alpha} \geq 0$ times continuously differentiable functions $f : \mathcal{X} \mapsto \mathbb{R}$ with $\|f\|_{\alpha,\infty} < r$. Due to smoothing, $v_N \in \mathbb{C}_r^\alpha(\mathcal{Z})$, for some $r < \infty$, if $u$ and $F_Z$ are sufficiently smooth (see Theorem 1). We can therefore import existing results for approximation methods for functions in $\mathbb{C}_r^\alpha(\mathcal{Z})$:

**Example 2.** Polynomial interpolation using tensor products. Suppose we use $J$th order Chebyshev interpolation with $M \geq J$ nodes in each of the $d_z$ dimensions, or a $J$th order B-spline interpolation with $M \geq J$ number of nodes in each of the $d_z$ dimensions (see Appendix C for their precise expressions). Let $p_1, ..., p_J$ denote the $J$ polynomials; we then have

$$B_K(z) = \left\{p_{j_1}(z_1) \cdots p_{j_{d_z}}(z_{d_z}) : j_1, ..., j_{d_z} = 1, ..., J\right\},$$

which is of dimension $K = J^{d_Z}$. Choosing $J \geq \underline{\alpha}$, where $\underline{\alpha} \geq 1$ denotes the number of derivatives of $v(z)$, both interpolation schemes satisfy, for any radius $r < \infty$,

$$\sup_{v \in \mathbb{C}_r^\alpha(\mathcal{Z})} \left\|\bar{\Pi}_K(v) - v\right\|_\infty = O\left(\frac{\log(J)}{J^\alpha}\right) = O\left(\frac{\log(K)}{K^{\alpha/d}}\right);$$

see p.14 in Rivlin (1990) for Chebyshev interpolation and Schumaker (2007) for B-splines. If

$v(z)$ is analytic ($\underline{\alpha} = \infty$), the above result holds for any (large) $J, \alpha < \infty$.

As can be seen from the above example, standard polynomial tensor product approximations suffer from the well-known computational curse of dimensionality: To reach a given level of error tolerance, the total number of basis functions $K$ has to grow exponentially as $d_Z$ increases. This issue can be partially resolved by using more advanced function approximation methods:

**Example 3.** Interpolation with sparse grids. Instead of using tensor-product basis functions to approximate a given function, where the total number of basis function and interpolation points will have to grow exponentially with $d_Z$ to control the approximation error, one can instead use so-called Smolyak sparse grids; see, e.g., Judd et al. (2014) and Brumm and Scheidegger (2017). Using these, the number of grid points needed to obtain a given error tolerance are reduced from $O\left(M^{d_Z}\right)$ to $O\left(M\left(\log M\right)^{d_Z}\right)$ with only slightly deteriorated accuracy.

**Example 4.** Variable selection, shape constraints, shrinkage estimators, and machine learning. An alternative way of breaking the curse of dimensionality appearing in Example 2 is to select the basis functions judiciously. This could, for example, be done using standard variable selection methods; one example of this approach can be found in Chen (1999). Alternatively, one can in some cases show that the value functions satisfy certain shape constraints that can then be imposed on the sieve; see, for example, Cai and Judd (2013). Other automated selection methods include shrinkage methods where a penalization term is added to the least-squares criterion. Again this leads to a more sparse representation which is able to break the curse-of-dimensionality. Finally, machine learning algorithms, such as neural networks, may potentially be useful in approximating the value functions; see, for example, Chen and White (1999). On the other hand, these methods are generally computationally more expensive compared to the least-squares projection method in (4.7) and require that the value function satisfies certain sparsity. We will investigate the performance of such more advanced projection operators in future work.

As noted earlier, there is no guarantee that a given function approximator is non-expansive. But this can, in principle, be examined numerically for a given choice of $\bar{\Pi}_K$. For the least-squares projection, this amounts to solving, for a given choice of basis functions and grid points,

$$\left\|\bar{\Pi}_K\right\|_{op,\infty} = \sup_{v\in\mathbb{R}^M, \|v\|=1} \sup_{z\in\mathcal{Z}} \left| B_K(z)' \left[\sum_{i=1}^{M} B_K(z_i) B_K(z_i)'\right]^{-1} \sum_{i=1}^{M} B_K(z_i) v_i \right|. \tag{4.9}$$

When $M$ and/or $\dim \mathcal{Z}$ is large this may be computationally demanding and instead one can obtain a lower bound by restricting $z$ to only take values on the chosen set of grid points: With $\mathbf{B}_{K,M} \in \mathbb{R}^{K\times M}$containing the basis functions evaluated at the grid points, we can represent $\bar{\Pi}_K$ when only evaluated at chosen grid points $z_1, ...., z_M$ in terms of

$$\mathbf{P}_{K,M} = \mathbf{B}'_{K,M}\left[\mathbf{B}_{K,M}\mathbf{B}'_{K,M}\right]^{-1}\mathbf{B}_{K,M} \in \mathbb{R}^{M\times M}.$$

In particular, it is easily checked that with the supremum in (4.9) being only taken over $z \in \{z_1, ..., z_M\}$, $\left\|\bar{\Pi}_K\right\|_{op,\infty} = \|\mathbf{P}_{\mathbf{K},\mathbf{M}}\|_{op,\infty}$. Furthermore, $\|\mathbf{P}_{K,M}\|_{op,\infty} \leq 1$ if and only if

$$\max_{i=1,...,M} \sum_{j=1}^{M} |p_{ij}| \leq 1,$$

where $p_{ij}$ is the $(i,j)$th element of $\mathbf{P}$, c.f. Lizotte (2011).

### 4.3.4 Solving for the approximate value functions

Computing the simulated self-approximating solution or the projection-based one can be done using three different numerical algorithms: Successive approximation (SA), Newton-Kantorovich (NK), or a combination of the two. The latter corresponds to the hybrid solution method proposed in Rust (1988). We here discuss the implementation of these algorithms with focus on the sieve-based approximation of $v$; the implementations of the sieve approximation of $V$ and the self-approximating solutions of either of the two follow along the same lines. The main difference between solving for $V$ or $v$ is that the latter involves smaller computational burden since it is a scalar function while the former is a $D$-dimensional vector function.

SA utilizes that (for $K$ chosen large enough), $\bar{\Pi}_K \bar{\Gamma}_{N,\lambda}$, is a contraction mapping which guarantees that the following algorithm will converge towards the solution to (4.2),

$$\hat{v}_N^{(k)} = \bar{\Pi}_K \bar{\Gamma}_N(\hat{v}_N^{(k-1)}), \tag{4.10}$$

for $k = 1, 2, ...$, given some initial guess $\hat{v}_N^{(0)}$. In the leading case of (4.7), this can be expressed as a sequence of least-squares problems that are easily computed: $\hat{v}_N^{(k)}(z) = \hat{\alpha}_k' B_K(z)$ where

$$\hat{\alpha}_k = \left[\sum_{i=1}^{M} B_K(z_i) B_K(z_i)'\right]^{-1} \sum_{i=1}^{M} B_K(z_i) \bar{\Gamma}_{N,\lambda}(\hat{\alpha}_{k-1}' B_K)(z_i)' \in \mathbb{R}^K,$$

for $k = 1, 2, ...$, given some initial guess $\hat{\alpha}_0$. In the case where $\varepsilon_t = \emptyset$ or when the model is on the form eq. (3.9) with $\varepsilon_t^{(1}$ being extreme-valued distributed and $\varepsilon_t^{(-1)} = \emptyset$,

$$\bar{\Gamma}_N(\alpha' B_K)(z_i; \lambda) = G_\lambda\left(u(z) + \beta\alpha' \sum_{j=1}^{N} B_K(Z_j(z_i)) w_{Z,N,j}(z_i)\right),$$

and so $\sum_{j=1}^{N} B_K(Z_j(z_i)) w_{z,N,j}(z_i)$, $i = 1, ..., M$, only need to be computed once and then recycled in each iteration; in contrast, the simulated averages appearing in $\Gamma_N(\hat{\alpha}_{k-1}' B_K)(z_i)$, $i = 1, ..., M$, have to be recomputed in each step of the SA algorithm. Thus, in this special case, it is faster to (approximately) solve for $v_{N,\lambda}$ instead of $V_{N,\lambda}$. While SA is guaranteed to converge globally when $\bar{\Pi}_K \bar{\Gamma}_N$ is a contraction, the rate of convergence will be slow with the

error vanishing at rate $\beta^k$,

$$\left\|\hat{v}_N^{(k)} - v_N\right\|_\infty \leq \frac{\beta^k (1+\beta)}{1-\beta} \left\|\hat{v}_N^{(0)} - v_N\right\|_\infty . \tag{4.11}$$

To speed up convergence, we therefore follow Rust (1988) and combine SA with NK iterations since NK converges with a quadratic rate once a given guess of the value function is close enough to the fixed point. Moreover, in situations where $\bar{\Pi}_K \bar{\Gamma}_N$ is expansive, NK is still guaranteed to converge locally. Since both the self-approximating and sieve-based methods solve finite-dimensional problems, the NK algorithm for these are equivalent to Newton's method. First consider the sieve-based method where we focus on the least-squares projection as given in (4.7). We are then seeking $\hat{\alpha}$ solving the following $K$ equations,

$$\bar{S}_{N,K}(\alpha;\lambda) = 0,$$

with

$$\bar{S}_{N,K}(\alpha;\lambda) = \alpha - \left[\sum_{i=1}^M B_K(z_i) B_K(z_i)'\right]^{-1} \sum_{i=1}^M B_K(z_i) \bar{\Gamma}_N(\alpha' B_K)(z_i;\lambda).$$

The corresponding derivatives of the left-hand side as a function w.r.t. $\alpha$ can be expressed in terms of the Hadamard differential of $\bar{\Gamma}_N$ w.r.t. $v$,

$$\begin{aligned} \nabla\bar{\Gamma}_N(v)[dv](z;\lambda) &= \beta \sum_{d\in\mathcal{D}} \sum_{j=1}^N \dot{G}_{d,\lambda}\left(u(z,\varepsilon_j(z)) + \beta \sum_{i=1}^N v(Z_i(z);\lambda) w_{Z,N,i}(z)\right) \\ &\times \left(\sum_{k=1}^N dv(Z_k(z,d);\lambda) w_{Z,N,k}(z,d)\right) w_{\varepsilon,N,j}(z), \end{aligned}$$

where $dv : \mathcal{Z} \times \left[0,\bar{\lambda}\right] \mapsto \mathbb{R}$ is the direction and

$$\dot{G}_{\lambda,d}^{(r)}(r) = \frac{\partial G_\lambda(r)}{\partial r(d)} = \frac{\exp\left(\frac{r(d)}{\lambda}\right)}{\sum_{d'\in\mathcal{D}} \exp\left(\frac{r(d')}{\lambda}\right)}. \tag{4.12}$$

The partial derivatives of $\bar{S}_{N,K}(\alpha;\lambda)$ then becomes

$$\bar{H}_{N,K}(\alpha;\lambda) = I_K - \left[\sum_{i=1}^M B_K(z_i) B_K(z_i)'\right]^{-1} \sum_{i=1}^M B_K(z_i) \nabla\bar{\Gamma}_N(\alpha' B_K)[B_K](z_i;\lambda)' \in \mathbb{R}^{K\times K}.$$

With these definitions, the NK algorithm takes the form

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} - \bar{H}_{N,K}^{-1}(\hat{\alpha}_{k-1};\lambda) \bar{S}_{N,K}(\hat{\alpha}_{k-1};\lambda).$$

The NK algorithm for the self-approximating method is on the same form, except that we now solve directly for the value function at the $N$ draws. With slight abuse of notation, let $v_N = \{v_{N,\lambda}(Z_k;\lambda) : k = 1,...,N\}$ be the vector of integrated values across the set of draws

solving $\bar{S}_N(v_N;\lambda)=0$ where

$$\bar{S}_{N,k}(v_N;\lambda)=v_{N,k}-\sum_{j=1}^{N}G_\lambda\left(u(Z_k,\varepsilon_j)+\beta\sum_{i=1}^{N}v_{N,i}w_{z,N,i}(Z_k)\right)w_{\varepsilon,N,j}(Z_k), \tag{4.13}$$

for $k=1,...,N$. The corresponding derivatives is $\bar{H}_N(\alpha;\lambda)=\left(\bar{H}_{N,1}(\alpha;\lambda),...,\bar{H}_{N,N}(\alpha;\lambda)\right)'\in\mathbb{R}^{N\times N}$ where, with $\mathbf{1}_N=(1,...,1)'\in\mathbb{R}^N$,

$$\bar{H}_{N,k}(\alpha;\lambda)=I_N-\nabla\bar{\Gamma}_N(v_N)[\mathbf{1}_N](z_k;\lambda)\in\mathbb{R}^N.$$

Finally, we note that the NK algorithm for the expected value function takes a similar form with the functional differential of $\Gamma_N$ given by

$$\nabla\Gamma_N(V)[dV](z;\lambda)=\beta\sum_{d\in\mathcal{D}}\sum_{i=1}^{N}\dot{G}^{(r)}_{\lambda,d}(r)\left(u(S_i(z))+\beta V(Z_i(z);\lambda)\right)dV(Z_i(z),d)\,w_{N,i}(z), \tag{4.14}$$

where $dV(z)=(dV(z,1),...,dV(z,D))'$.

Comparing the NK algorithm for the self-approximating and the sieve-based method, we note that the former involves inverting a $N\times N$-matrix while the latter a $K\times K$-matrix. As pointed out earlier, the self-approximating method generally needs $N$ to be chosen quite large to achieve a precise simulated version of the Bellman operator, in particular in higher dimensions, and so the NK algorithm for this method may become numerically infeasible in some cases. While the projection-based method also suffers from a curse of dimensionality, since the number of basis functions, $K$, has to be quite large in higher dimensions to achieve a reasonable approximation, it is less severe and is implementable for higher-dimensional models. If more advanced function approximation methods are employed, even better performance can be achieved.

# 5 Theory

We here develop an asymptotic theory for the self-approximating and sieve-based methods. We first establish some important properties of the smoothed simulated Bellman operators and their exact solutions, $v_N$ and $V_N$ defined in (4.1). These are then used in the asymptotic analysis of the self-approximating solution method and the sieve-based one. This analysis will rely on two general results for estimated solutions to fixed point problems as stated in Theorems A.1 and A.2 in the appendix. The asymptotic analysis will mostly focus on $V_N$ and $\hat{V}_N$ since our results for these easily translate into similar results for the approximate integrated value function. For example, $v_N(z;\lambda)=M_{N,u}(\beta V_N(z;\lambda)|z;\lambda)$, where

$$M_{N,u}(r|z;\lambda)=\sum_{j=1}^{N}G_\lambda(u(z,\varepsilon_j(z))+r)\,w_{\varepsilon,N,j}(z),$$

and so the asymptotic results for $V_N$ in conjunction with the functional Delta method can be used to obtain similar results for $v_N$.

Without loss of generality, we assume that the draws can be written as

$$Z_i(z) = \psi_Z(U_i; z) \in \mathcal{Z},\ \varepsilon_i(z) = \psi_\varepsilon(U_i; z) \in \mathcal{E}, \tag{5.1}$$

for some i.i.d. draws $U_i \sim P_U$, $i = 1, ..., N$, and some functions $\psi_Z$ and $\psi_\varepsilon$. We then define, with $\psi = (\psi_Z, \psi_\varepsilon)$ and for any given function $V(z, \lambda)$,

$$u_\psi(U; z) := u(\psi(U; z)),\ w_\psi(U; z) = w(\psi(U; z)|z),\ V_\psi(U; z, \lambda) = V(\psi_Z(u; z), \lambda) \tag{5.2}$$

so that

$$\Gamma(V)(z, \lambda) = E\left[G_\lambda(u_\psi(U; z) + \beta V_\psi(U; z, \lambda))\, w_\psi(U; z)\right] \tag{5.3}$$

where expectations are taking over $U \sim P_U$, and

$$\Gamma_N(V)(z, \lambda) = \sum_{i=1}^{N} G_\lambda(u_\psi(U_i; z) + \beta V_\psi(U_i; z, \lambda))\, w_{\psi,N}(U_i; z), \tag{5.4}$$

where $w_{\psi,N}(U_i; z, d) = w_\psi(U_i; z, d) / \sum_{j=1}^{N} w_\psi(U_j; z, d)$. Here, and in the following, we let $V_0(z, \lambda)$ denote the exact solution to

$$V_0(z, \lambda) = \Gamma(V_0)(z, \lambda) \tag{5.5}$$

As explained earlier, we here define the two operators to take a given function $V(z, \lambda)$, $(z, \lambda) \in \mathcal{Z} \times \left[0, \bar{\lambda}\right]$, for some given $\bar{\lambda} > 0$, and map them into another function with domain $\mathcal{Z} \times \left[0, \bar{\lambda}\right]$. In particular, $V_0(z, 0)$ and $V_N(z, 0)$ are the non-smoothed ($\lambda = 0$) exact and simulated solutions, respectively. We then impose the following regularity conditions on the model and chosen importance sampler, where we recall the function norm defined in eq. (4.8) and the function set $\mathbb{C}_r^\alpha(\mathcal{Z})$ defined below this equation:

**Assumption 1.** *The support $\mathcal{Z}$ is a compact set; $\bar{u}_\psi(u) := \sup_{z \in \mathcal{Z}} \|u_\psi(u; z)\|$ and $\bar{w}_\psi(u) := \sup_{z \in \mathcal{Z}} w_\psi(u; z)$ satisfy $E\left[\bar{u}_\psi^2(U)\, \bar{w}_\psi^2(U)\right] < \infty$.*

**Assumption 2.** *For some $\alpha > 0$, $z \mapsto u_\psi(U; \cdot)$ and $z \mapsto w_\psi(U; z)$ belong to $\mathbb{C}_\infty^\alpha(\mathcal{Z})$ $P_U$-almost surely with $E\left[\|u_\psi(U; \cdot)\|_{\alpha,\infty}^2 \|w_\psi(U; \cdot)\|_{\alpha,\infty}^2\right] < \infty$.*

Assumption 1 share some similarities with the regularity conditions found in Rust (1988) who considered an additive version of our general model. Importantly, we only require that $Z_t$ has bounded support while $\varepsilon_t$ can have potentially unbounded support. This is in contrast to Pal and Stachurski, 2013 and Rust, 1997b who require both components to be bounded. We conjecture that the subsequent results can be generalized to also hold in the case of $\mathcal{Z}$ unbounded but then our conditions and arguments would have to be changed. For example, the existence of unique fixed points would have to be verified in a function space equipped

with a weighted sup-norm and with additional moment conditions on $Z_t$, see, e.g., Norets (2010). Similarly, our empirical process results would need to be established using bracketing conditions with weighted norms and additional moment conditions, see, e.g., Section 2.10.4 in van der Vaart and Wellner, 1996. Similar to the results in Rust (1988), Assumption 1 implies $\Gamma(V) \in \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ for all $V \in \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$, see below. This particular result actually holds under the weaker requirement that $E\left[\|u_\psi\left(U;\cdot\right)\|_{0,\infty}\|w_\psi\left(U;\cdot\right)\|_{,\infty}\right] < \infty$ but the existence of the second order moment is needed for the subsequent asymptotic analysis of $V_N\left(z;\lambda\right)$ and so we impose this restriction throughout.

Assumption 2 impose smoothness conditions on the model and the chosen samplers in terms of the state variables $Z_t$. These conditions imply that the expected and integrated value functions will be smooth too. Note here that the degree of smoothness $\alpha$ is left unrestricted at this stage and so the functions are not required to be differentiable, merely Lipschitz. While the focus is on models with continuous state variables our theory also covers models with discrete state space. In this case, we can dispense of Assumption 2 and instead rely on the more general Theorem B.1 which implies that the subsequent results still go through when $Z_t$ is discrete.

We first establish existence and uniqueness of the (generally) infeasible simulated solutions and show that they inherit the smoothness properties of $z \mapsto u_\psi\left(U;\cdot\right)$ and $z \mapsto w_\psi\left(U;z\right)$. This feature of the approximate solutions is important for two reasons: First, it allows us to show uniform convergence of certain functionals as part of our proof of weak convergence. Second, we can control the approximation error due to the use of sieves later on.

**Theorem 1.** *Suppose Assumption 1 holds and, for a given $N \geq 1$, $\inf_{z\in\mathcal{Z}} \sum_{i=1}^N w_\psi\left(U_i;z\right) > 0$. Then the operators $\Gamma$ and $\Gamma_N$ in eqs. (5.3) and (5.4) are almost surely contraction mappings on $\mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ and so $V_0 : \mathcal{Z} \times \left(0, \bar{\lambda}\right) \mapsto \mathbb{R}^D$ and $V_N : \mathcal{Z} \times \left(0, \bar{\lambda}\right) \mapsto \mathbb{R}^D$ exist and are unique. If furthermore Assumption 2 holds then $V_0 \in \mathbb{C}^\alpha_{r_0}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)$ for some constant $r_0 < \infty$ while $V_N \in \mathbb{C}^\alpha_{r_N}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)$ for some $r_N < \infty$ $P_U$-almost surely.*

The above is a fixed $N$ result with the bound on $V_N$, $r_N$, being random since it depends on the particular set of draws. We derive a deterministic bound on $r_N$ as $N \to \infty$ below. The condition $\inf_{z\in\mathcal{Z}} \sum_{i=1}^N w_\psi\left(U_i;z\right) > 0$ will hold with probability approaching 1 (w.p.a.1) as $N \to \infty$ and so can be dropped in our asymptotic analysis. Next, we analyze the effect of smoothing on the exact and simulated value function:

**Theorem 2.** *Under the conditions of Theorem 1, the following hold: $\left\|V_0\left(\cdot;\lambda\right) - V_0\left(\cdot;0\right)\right\|_\infty = O\left(\lambda\right)$ and $\left\|V_N\left(\cdot;\lambda\right) - V_N\left(\cdot;0\right)\right\|_\infty = O_P\left(\lambda\right)$ for any given $N \geq 1$.*

This shows that the smoothing can be controlled for by suitable choice of $\lambda$ both asymptotically ($N = +\infty$) and for any finite number of simulations ($N < \infty$). Also note that the above result holds independently of the smoothness properties of the unsmoothed exact and simulated solutions.

## 5.1 Self-approximating method

In this section, we provide an analysis of the smoothed fixed point, $V_N$, to $\Gamma_N$ defined in eq. (5.4) thereby allowing for general importance samplers. As a special case, we obtain an asymptotic theory for the self-approximating solution method (where $\Phi_z(z'|z,d)$ is restricted to be marginal distribution). The results for $V_N$ will then in turn be used in the analysis of the corresponding sieve-based methods in the next section.

**Theorem 3.** *Suppose Assumptions 1-2 hold for some $\alpha > 0$. Then $V_N$ solving $\Gamma_N(V_N) = V_N$ satisfies $\|V_N - V_0\|_\infty = O_P(1/\sqrt{N})$.*

*If Assumption 2 holds with $\alpha \geq 1$, then $V_0, V_N \in \mathbb{C}^1_r\left(\mathcal{Z} \times \left[0, \bar{\lambda}\right]\right)$ w.p.a.1 for some constant $r < \infty$. Moreover, $\sup_{z\in\mathcal{Z}} \|\partial V_N(z,\lambda)/(\partial z_j) - \partial V_0(z,\lambda)/(\partial z_j)\| = O_P\left(\sqrt{N}/\lambda\right)$ uniformly over $\lambda \in \left(0, \bar{\lambda}\right)$.*

The first part of this theorem is similar to results found in Rust (1997b) and Pal and Stachurski (2013) who also show $\sqrt{N}$-convergence of their value function approximation. Importantly, the convergence result holds uniformly over the smoothing parameter $\lambda$ and so there is no first-order effect from smoothing if $\lambda$ vanishes sufficiently fast. Specifically, for any sequence $\lambda_N$ satisfying $\sqrt{N}\lambda_N \to 0$, Theorems 2 and 3 yield $\sup_{z\in\mathcal{Z}} \|V_N(z,\lambda_N) - V(z,0)\| = O_p(1/\sqrt{N})$. This is similar to convergence of smoothed empirical cdf where the indicator function is replaced by a smoothed version; this also does not affect the convergence rate as long as the smoothing bias is controlled for. The second part of the theorem appears to be a new result and shows that if the problem is smooth enough, the first-order partial derivatives of $V_N(z,\lambda)$ also converge uniformly over $z$ with rate $\sqrt{N}/\lambda$. Since we need $\lambda \to 0$ to kill the smoothing bias, this could seem to imply that the first-order derivatives converge with slower than $\sqrt{N}$-rate. However, we conjecture that the derived rate is not sharp and that $\sqrt{N}$-convergence does actually hold. The proof of this appears to require a more delicate and refined arguments, however, and so we leave this for future research.

The above result is then in turn used to derive the asymptotic distribution of $V_N(z,\lambda)$ uniformly in $(z,\lambda) \in \mathcal{Z} \times \left(0, \bar{\lambda}\right)$. Here, the smoothing proves important since it allows us to generalize the standard arguments used in the analysis of finite-dimensional extremum estimators to our setting: We first expand the "first-order condition", $V_N - \Gamma_N(V_N) = 0$, around $V_0 = \Gamma(V_0)$ to obtain, with $\nabla\Gamma_N$ defined in (4.14),

$$0 \;=\; \Gamma(V_0) - \Gamma_N(V_0) + \{I - \nabla\Gamma_N(V_0)\}[V_N - V_0] + o_P\left(1/\sqrt{N}\right),$$

where the rate of the remainder term follows from Theorem 3. Next, employing empirical process theory, we show that $\sqrt{N}\{\Gamma(V_0) - \Gamma_N(V_0)\} \rightsquigarrow \mathbb{G}$ in $\mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ for a Gaussian process $\mathbb{G}(z,\lambda)$ with covariance kernel

$$\Omega(z_1,\lambda_1,z_2,\lambda_2) = E_U\left[g(U;z_1,\lambda_1)\, g(U;z_2,\lambda_2)'\right], \tag{5.6}$$

$$g\left(U;z,\lambda\right)=\left\{G_{\lambda}\left(u_{\psi}\left(U;z\right)+\beta V_{0}\left(\psi_{Z}\left(U;z\right),\lambda\right)\right)-\Gamma(V_{0})\left(z,\lambda\right)\right\}w_{\psi}\left(U;z\right). \tag{5.7}$$

Finally, we show that $\nabla\Gamma_{N}(V_{0})\left[dV\right]\to^{P}\nabla\Gamma(V_{0})\left[dV\right]$ uniformly over $dV\in\mathbb{C}_{2r}^{1}\left(\mathcal{Z}\times\left[0,\bar{\lambda}\right]\right)^{D}$ with $r<\infty$ given in Theorem 3. Since $V_{N}-V_{0}\in\mathbb{C}_{2r}^{1}\left(\mathcal{Z}\times\left[0,\bar{\lambda}\right]\right)^{D}$, we conclude that:

**Theorem 4.** *Suppose Assumptions 1 and 2 hold with $\alpha\geq 1$. Then, $\sqrt{N}\{V_{N}-V_{0}\}\rightsquigarrow\mathbb{G}_{V}$ on $\mathbb{B}\left(\mathcal{Z}\times\left(0,\bar{\lambda}\right)\right)^{D}$ where $\mathbb{G}_{V}\left(z,\lambda\right)=\left\{I-\nabla\Gamma(V_{0})\right\}^{-1}\left[\mathbb{G}\right]\left(z,\lambda\right)$ is a $D$-dimensional Gaussian process.*

The above result implies, for example, that $\sqrt{N}\left\{V_{N}\left(z,\lambda\right)-V_{0}\left(z,\lambda\right)\right\}\to^{d}N\left(,\Omega_{V}\left(z,\lambda,z,\lambda\right)/N\right)$ as $N\to\infty$ for any given $(z,\lambda)$, where

$$\Omega_{V}\left(z_{1},\lambda_{1},z_{2},\lambda_{2}\right)=\int\int r^{*}\left(z_{1}',\lambda_{1}'|z_{1},\lambda_{1}\right)\Omega\left(z_{1}',\lambda_{1}',z_{2},\lambda_{2}\right)r^{*}\left(z_{2}',\lambda_{2}'|z_{2},\lambda_{2}\right)'d\left(z_{1}',\lambda_{1}'\right)d\left(z_{2}',\lambda_{2}'\right),$$

where $r^{*}$ is the Riesz representer of $dV\mapsto\left\{I-\nabla\Gamma(V_{0})\right\}^{-1}\left[dV\right]\left(z,\lambda\right)$. Thus, it allows us to construct (pointwise or uniform) confidence bands for the expected value function. We expect that the result will also be useful in analyzing the impact of value function approximation when used in estimation. This could be done by combining the above weak convergence result with, e.g., the results for approximate estimators found in Kristensen and Salanie (2017).

The proof of Theorem 4 proceeds by verifying the two high-level conditions of the "master" Theorem B.1 where the same weak convergence result is obtained under more general conditions. Theorem B.1 allows us to replace the smoothness conditions in Assumption 2 with some other conditions implying that $V_{N}-V_{0}$ is situated in a function set with finite entropy. One example would be to impose restrictions on $u$ and $w$ so that the value function and its estimator are both monotone functions, c.f. Pal and Stachurski, 2013, in which case we could then appeal to Theorem 2.7.5 in van der Vaart and Wellner, 1996 to obtain the results of Theorem 4.

We conjecture that a similar weak convergence result will hold for the non-smoothed value function approximation ($\lambda=0$). However, the proof of such a result would require different arguments and seemingly stronger assumptions. In particular, the current proof only requires the empirical process $(z,\lambda)\mapsto\Gamma_{N}(V_{0})\left(z,\lambda\right)$ to converge weakly. To allow for non-smooth value function approximation, we conjecture that we would now need to show that the empirical process $(V,z)\mapsto\Gamma_{N}(V)\left(z,0\right)$ converges weakly over a suitable function set that the estimated non-smooth solution, $V_{N}$, would be situated in. For this to hold, the uniform entropy of the function set would need to be finite. Standard choices of function sets are smooth classes, but $V_{N}$ and its limit $V_{0}$ are both non-smooth now and so the proof appears to be rather delicate.

Finally, for a complete analysis that takes into account the smoothing bias, we state the following corollary to Theorem 4: For any $\lambda_{N}\to 0$ such that $\lambda_{N}\sqrt{N}\to 0$, $\sqrt{N}\{V_{N}\left(\cdot;\lambda_{N}\right)-V_{0}\left(\cdot;0\right)\}\rightsquigarrow\mathbb{G}_{V}\left(\cdot,0\right)$.

## 5.2 Sieve-based approximation of value functions

We now proceed to analyze the asymptotic properties of the sieve-based approximate value function, $\hat{V}_N$. To this end, we use the following decomposition of the over-all error,

$$\hat{V}_N - V_0 = \left\{\hat{V}_N - V_N\right\} + \left\{V_N - V_0\right\}, \tag{5.8}$$

where the second term converges weakly towards a Gaussian process, c.f. Theorem 4. What remains is to control the first term which is due to the sieve approximation; this is done by imposing the following high-level assumption on the projection operator when applied to a function set $\mathcal{V}$ which is chosen so that $V_N \in \mathcal{V}$ w.p.a.1.:

**Assumption 3.** *The projection operator $\Pi_K$ satisfies $\sup_{v \in \mathbb{C}^\alpha_{r_0}(\mathcal{Z})} \|\Pi_K(v) - v\|_\infty = O_P(\rho_K)$ for some sequence $\rho_K \to 0$, where $\alpha$ is given in Assumption 2 and $r_0 < \infty$ in Theorem 1.*

This is a high-level condition that requires the chosen function approximation method to have a uniform error rate over the function class $\mathbb{C}^\alpha_r(\mathcal{Z})$ which we know $z \mapsto V_0(z, \lambda)$ belongs uniformly in $\lambda$ under Assumption 2. As discussed earlier, one could replace Assumption 2 with other regularity conditions that ensure $z \mapsto V_0(z, \lambda)$ is sufficiently regular (e.g., monotonic) in which case $\mathbb{C}^\alpha_r(\mathcal{Z})$ in Assumption 3 should be modified accordingly. Assumption 3 is satisfied for standard polynomial approximators with $\rho_K = \log(K)/K^{(s+1)/d}$, c.f. Section 4.3. Compared to results on sieve approximations of value functions found elsewhere in the literature, our rate is better since we are here seeking to approximate the expected value function that is situated in $\mathbb{C}^\alpha_r(\mathcal{Z})$. In contrast, sieve-based approximations developed in other papers, such as Munos and Szepesvari (2008) and Pal and Stachurski (2013), try to approximate the value function which is at most Lipschitz and for such functions the approximation error will be larger in general. In the case of $\mathcal{Z}$ being finite, we have $\sup_{V \in \mathcal{V}} \|\Pi_K(V) - V\|_\infty = 0$ for $K > |\mathcal{Z}|$ under great generality and so there will be no asymptotic bias component due to sieve approximations in this case.

The second part of Theorem A.1 together with the fact that $\Gamma_{N,\lambda}(V_\lambda) - \Gamma_\lambda(V_\lambda) = O_P\left(1/\sqrt{N}\right)$, c.f. Proof of Theorem 3, now yield the following result:

**Theorem 5.** *Suppose Assumptions 1-3 hold. Then $\hat{V}_N$, defined as the solution to $\Pi_K \Gamma_N(\hat{V}_N) = \hat{V}_N$, satisfies $\|\hat{V}_N - V_0\|_\infty = O_p(1/\sqrt{N}) + O_P(\rho_K)$. Suppose in addition that $\alpha \geq 1$ in Assumption 2. Then, if $\sqrt{N}\rho_K \to 0$, $\sqrt{N}\{\hat{V}_N - V_0\} \rightsquigarrow \mathbb{G}_V$.*

The discussions following Theorems 3 and 4 carry over to the above result. In particular, the rate result still goes through when no smoothing is employed ($\lambda = 0$) but the current proof of the asymptotic distribution result requires smoothing ($\lambda > 0$). Compared to the rate results for $V_{N,\lambda}$, the projection-based method suffers from an additional error due to the sieve approximation, $O_P(\rho_K)$. This can be interpreted as a bias term, while $O_p(1/\sqrt{N})$ is its variance component which is shared with $V_{N,\lambda}$. The requirement that $\sqrt{N}\rho_K \to 0$ is used to kill the sieve bias term so that $\hat{V}_N$ is centered around $V_0$.

The above result provides a refinement over existing results where a precise rate for the bias is not available; see, e.g., Lemma 5.2 in Pal and Stachurski (2013). It shows that there is an inherent computational curse-of-dimensionality built into our projection-based value function approximation when polynomial interpolation is employed: In high-dimensional models, a large number of basis functions are needed which in turn increases the computational effort. In the case of polynomial approximations, the rate condition becomes $\sqrt{N}\log(K)/K^{(s+1)/d} \to 0$ and so, as $d$ increases, we need $K$ to increase faster with $N$ to kill the sieve bias component. However, also note that $K$ as no first-order effect on the variance and so there is no bias-variance trade-off present. In particular, we can let $K$ increase with $N$ as fast as we wish and so our procedure should in principle also work for models with high-dimensional state space. That is, we can achieve $\sqrt{N}$-rate regardless of the dimension of the problem and so our method does not suffer from any statistical curse-of-dimensionality. However, this requires choosing $K$ large enough in order to control the sieve approximation bias which will increase computation time as the dimension grows. Thus, there is a potential computational curse-of-dimensionality.

# 6 Numerical results

In this section we examine the numerical performance of the proposed solution algorithms with focus on how the theoretical results derived in the previous sections translate into practice and how different features of model and implementation affect their performances.

We focus exclusively on approximating the integrated value function, $v(z)$, and measure the performance of a given approximate solution, say, $\tilde{v}(z)$ in terms of its pointwise bias, variance and mean-square error (MSE) defined as $Bias(z) := E[\tilde{v}(z)] - v(z)$, $Var(z) := Var(\tilde{v}(z)) = E[(\tilde{v}(z) - E[\tilde{v}(z)])^2]$ and $MSE(z) = Bias^2(z) + Var(z)$, respectively. As overall measures we use uniform bias, variance and MSE, $\|Bias\|_\infty = \sup_{z\in\mathcal{Z}} |Bias(z)|$, $\|Var\|_\infty = \sup_{z\in\mathcal{Z}} |Var(z)|$ and $\|MSE\|_\infty = \sup_{z\in\mathcal{Z}} |MSE(z)|$. Given that the exact solution $v(z)$ is unknown, we replace this by a very precise approximate solution computed in the following way: First, instead of using simulations in the computation of the Bellman operator, we utilize that the state transitions follow a Beta distribution in the chosen model (see below) and so we can use nodes and weights based on Jacobi polynomials to compute it using numerical integration. We then implement the sieve method using $K = 60$ Chebyshev polynomials and $N = 60$ sets of quadrature nodes and weights. The "exact" solution was computed by successive approximation until a contraction tolerance of machine precision was reached. We approximate the pointwise bias and variance of a given method through $S \geq 1$ independent replications of it: Let $\tilde{v}_1(z), ...., \tilde{v}_S(z)$ be the solutions obtained across the $S$ replications, where $S$ generally was chosen to 2,000. We then approximate the mean by $\hat{E}[\tilde{v}(z)] = \frac{1}{S}\sum_{s=1}^{S} \tilde{v}_s(z)$ which in turn is used to obtain the following pointwise bias and variance estimates, $\hat{Bias}(z) = \hat{E}[\tilde{v}(z)] - v_0(z)$, and $\hat{Var}(z) = \frac{1}{S}\sum_{s=1}^{S}(\tilde{v}_s(z) - \hat{E}[\tilde{v}(z)])^2$. Based on these, we approximate $\|Bias\|_\infty$ and $\|Var\|_\infty$ by the maximum pointwise biases and variances over a uniform grid over $[0, 1000]$ of size 500 in the univariate case and a uniform grid over $[0, 1000]^2$ of size 250.

To implement the sieve-based method, we need to choose the sieve space used in constructing $\Pi_K$. We here focus on Chebyshev basis functions and B-Splines as discussed in Section 4.3[1].

## 6.1 A model of optimal replacement

To provide a test bed for comparison of the sieve-based approximation method, we use the well-known engine replacement model by Rust (1987). Rust's model has become the basic framework for modeling dynamic discrete-choice problems and has been extensively used in other studies to evaluate the performance of alternative solution algorithms and estimators. While the model and its solution is well described in many papers, for completeness we briefly describe our variation of it below.

We consider the optimal replacement of a durable asset (such as a bus engine) whose controlled state $Z_t \in \mathbb{R}_+$ is summarized by the accumulated utilization (mileage) since last replacement. In each period, the decision maker faces the binary decision $d_t \in \mathcal{D} = \{0, 1\}$ whether to keep ($d_t = 0$) or replace ($d_t = 1$) the durable asset with a fixed replacement cost $RC > 0$. If the asset is replaced, accumulated usage $Z_t$ regenerates to zero. The maintenance/operating costs are assumed to be linear in usage $Z_t$, $c(Z_t) = \theta_c \cdot 0.001 \cdot Z_t$. The state and decision dependent per period utility is then given by $\bar{u}(Z_t, d_t) + \varepsilon_t(d_t)$ where $\bar{u}(Z_t, d_t) = (RC + c(0))\, I\{d_t = 0\} + c(Z_t)\, I\{d_t = 1\}$ and the utility shocks $\varepsilon_t = (\varepsilon_t(0), \varepsilon_t(1))$ are i.i.d. extreme value and fully independent of $Z_t$. This specification is a special case of Example 1 with $\lambda = \sigma_\varepsilon$ and so the simulated Bellman operator takes the form (3.5) where $G_{\sigma_\varepsilon}(\cdot)$ appears as part of the model. Thus, there is no smoothing bias present in the baseline model. In Section 6.5, we investigate the effect of smoothing by pretending that we are not able to integrate out $\varepsilon_t$ analytically in the baseline model and instead we simulate both $Z_t$ and $\varepsilon_t$ and then include our smoothing device in the computation of the simulated Bellman operator.

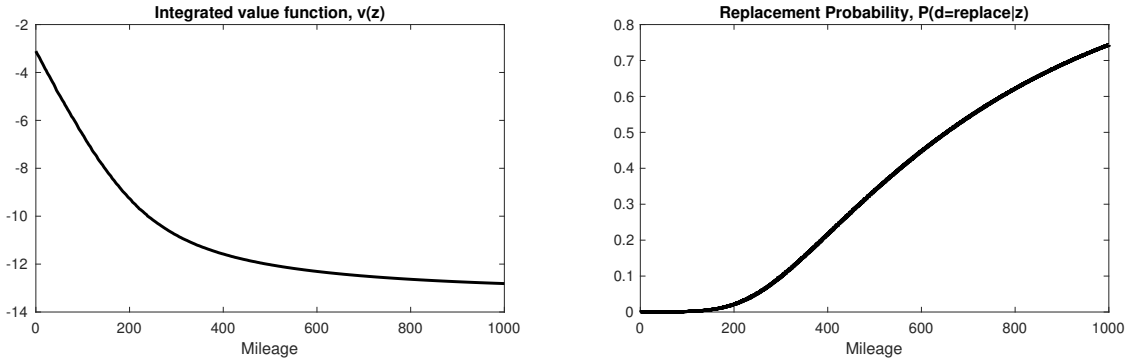
We assume that $Z_t$ (in absence of the replacement decision) follows a mixture of a discrete distribution with a probability mass $\pi > 0$ at zero and a linearly transformed Beta distribution with shape parameters $a$ and $b$ and scale parameter $\sigma_\varepsilon > 0$. Thus,

$$F_Z(z'|z, d) = \pi I\{z' = z\} + (1 - \pi) F_+(z'|z, d), \tag{6.1}$$

where $F_+(z'|z, d)$ has density $f_+(z'|z, d) = f_\beta((z' - z)/\sigma_Z; a, b)/\sigma_Z$, $\pi > 0$ is the probability of no usage and $f_\beta(x; a, b)$ is the probability density function of the Beta distribution with shape parameters $a, b$. Note here that it has bounded support $(0, \sigma_Z)$ so that $f_+(z'|z, d) = 0$ for $z' < z$ or $z' - z > \sigma_Z$. This is in line with the discretized model in the original formulation in Rust (1987) where monthly mileage were only allowed to take a few discrete values and monthly mileage is naturally bounded above and below (busses never drives backwards and there are limits how far a bus can drive within a month). We introduce probability mass $\pi$ at $z' = z$ to

[1] For more details on their implementation, see appendix C .

Figure 1: Fine Approximation as "Exact" Solution




Notes: Discount factor is $\beta = 0.95$, utility function parameters are $\theta_c = 2$, $RC = 1$, $\lambda = 1$ and transition parameters are $\sigma_Z = 15$, $a = 2$, $b = 5$ and $\pi = 0.000000001$.

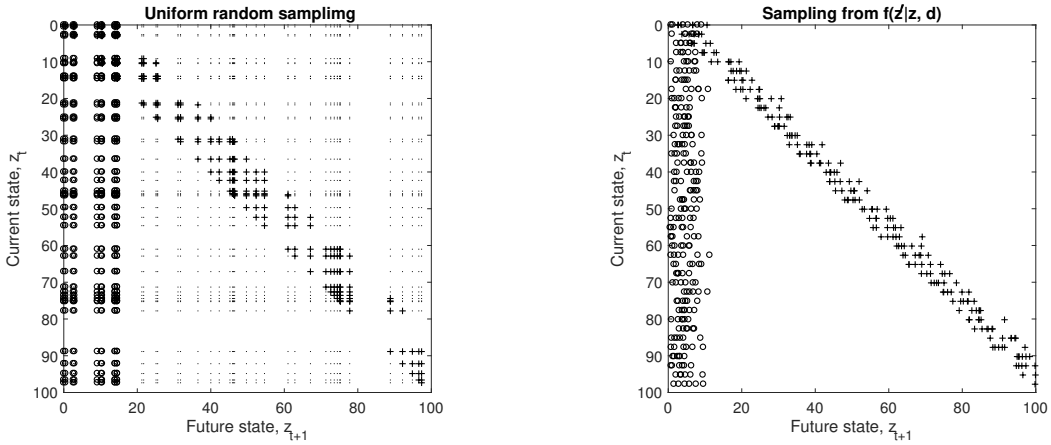
allow for the possibility that the asset is not used in a given period and thereby can end in the same state with positive probability when $\pi > 0$. As explained below, this feature turns out to be quite important for the applicability of the self-approximating method of Rust (1997). Note that the support of $Z_t$ is unbounded (the positive half line) and therefore the theory does not apply directly, since we throughout assumed bounded support. However, we expect that the theory extends to the unbounded case after suitable modifications, c.f. discussion following Assumptions 1-2.

In the numerical illustrations below we use the following set of benchmark parameter values unless otherwise specified: We set replacement cost to $RC = 10$ and the cost function parameter to $\theta_c = 2$ so that $RC$ is 5 times as large as $c(1000)$. This implies a large variation in the probability of replacement over $Z_t$ compared to Rust (1987) and a more curved value function. The parameters indexing the transition density $f_+(z'|z,d)$ are set to $\sigma_Z = 15$, $a = 2$, $b = 5$ and $\pi = 10^{-10}$ as default. This implies a quite sparse transition density, which is similar to the fitted model in Rust (1987). In Figure 1 we plot the corresponding "exact" solution as described earlier. Importantly, since the transition density is an analytic function the value function is also analytic and so well-approximated by polynomial interpolation methods.

## 6.2 Numerical implementation of simulated Bellman operators

The simulated Bellman operators in (3.6) and (3.7) require the user to choose an importance sampling distribution. For the self-approximating solution method we need to choose a marginal sampler, $d\Phi_Z(z'|z,d) = \phi_Z(z')\,dz'$. We follow Rust (1997b) and choose $\phi_Z(z') = I\{0 < z' < z^{\max}\}$ as a uniform density with support support $[0, z^{\max}]$ for some truncation point $0 < z^{\max} < \infty$ chosen by us. First note that this entails that the simulated Bellman operator used for the self-approximating value function is biased since we do not sample from the full support $\mathcal{Z} = \mathbb{R}_+$; however, this bias can be controlled by choosing $z^{\max}$ large enough. We will explain below why we do not choose $\phi_Z(z')$ as a density with support $\mathbb{R}_+$. Using a uniform sampler, the corresponding Radon-Nikodym derivative takes the form

Figure 2: Random Grids



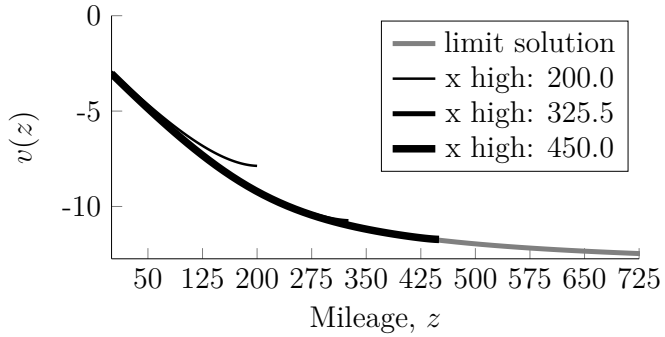

Notes: In the left panel we present the grids used for the self-approximate random Bellman operator. We have uniformly sampled a random grid, $\{Z_1, ..., Z_N\}$ on the interval $[0; 1000]$ with $N$ =400. Dots (.) mark sampled grid points in $R^2$: $Z_N \times Z_N$, plus (+) mark grid points where $f(z_j|z_i, d = 0) > 0$ and circles (o) mark points where $f(z_j|z_i, d = 1) > 0$. In the right panel, we plot the grid the projected random Bellman operator, where we have sampled directly from the conditional transition density in each of the $M = 400$ uniformly spaced evaluation points. To have equally many grid-points with non-zero transition density we only need $N = 400 * \sigma_Z / \max(Z_N) = 9$ random grids for each of the $M = 400$ evaluation points. Both figures show only a subset of the state space, $(z, z') \in [0; 100]^2$. Parameters are $\sigma_Z = 15$, $a = 2$, $b = 5$ and $\pi = 0.0000000001$.

$w_Z(z'|z, d) = \pi\delta(z' - z) + (1 - \pi)f_+(z'|z, d)$ where $\delta(\cdot)$ denotes Dirac's delta function. We approximate this by $\hat{w}_Z(z'|z, d) = \pi I\{z' = z\} + (1 - \pi)f_+(z'|z, d)$ which entails another small approximation error. For the sieve-based version, we simply choose $\Phi_Z(z'|z, d) = F_Z(z'|z, d)$ and so $w_Z(z'|z, d) = 1$.

As explained in Section 4.3, using a marginal importance sampler creates issues since it fails to adapt to the particular shape of the support of $F_Z(z'|z, d)$. In particular, for a given choice of $z$, many of the draws from $\phi_Z(z')$ will tend to fall outside the support of $f_Z(z'|z, d)$ and so will not contribute. In contrast, when $\phi_Z(z'|z, d) = f_Z(z'|z, d)$, the draws from $\phi_Z$ will by construction fall within the support of $f_Z(z'|z, d)$. This can be seen in Figure 2 where we have plotted the random draws obtained from the two different importance samplers used for the sieve-based and self-approximating solutions together with the actual support of $f_Z(z'|z, d)$. In the left-hand side panel we have plotted pairs of the uniform draws, $(Z_i, Z_j)$ for $i, j = 1, ..., N$, used for Rust's self-approximating method with $N = 400$ and $z^{max} = 1,000$, while in the right-hand side we have plotted $(z_i, Z_j(z_i, d))$ where $z_i$ are uniform draws and $Z_j(z, d) \sim f_Z(\cdot|z, d)$. In both cases, we have marked the pairs for which the corresponding density, $f_Z(Z_j|Z_i, d)$ and $f_Z(Z_j(z_i, d)|z_i, d)$, respectively, is positive. Clearly, the use of a marginal importance sampling density leads to very poor coverage of the actual support of $f_Z(z'|z, d)$ as $z$ varies while by construction $\Phi_Z(z'|z, d) = F_Z(z'|z, d)$ does an excellent job. This translates into the former simulated Bellman operator exhibiting much larger variance compared to the latter.

This issue is further amplified when we introduce the normalization given in eq. (3.3): Suppose that we had not included a discrete component $\pi I\{z' = z\}$ in the model. Then, with

Figure 3: Truncation bias due to $z^{max}$ being too low.




$Z_i \sim U[0, z^{\max}]$, $w_{N,Z,i}(Z_j, d) = f_+(Z_i|Z_j, d) / \sum_{k=1}^N f_+(Z_k|Z_j, d)$. Since $f_+(z'|z, d)$ has bounded support, it often happens that $\sum_{k=1}^N f_+(Z_k|Z_j, d) = 0$ for even large values of $N$ and so the simulated Bellman operator is not even well-defined. This issue will vanish as $N \to \infty$, but this on the other hand increases the computational burden since the self–approximating method require us to solve for the value function at the $N$ draws. Introducing the discrete component in the model resolves this issue since now $w_{Z,N,i}(Z_j, d) = \hat{w}_Z(Z_i|Z_j, d) / \sum_{k=1}^N \hat{w}_Z(Z_k|Z_j, d)$, where $\sum_{k=1}^N \hat{w}_Z(Z_k|Z_j, d) > 0$ for all $j = 1, ..., N$ by construction. Thus, $\pi > 0$ functions as a regularization device.

Why not choose $\phi_Z(z)$ as a density with unbounded support in order to avoid the issue of truncation? In our initial experimentation, we did try out sampling from distributions with unbounded support, but the above numerical issues became even more severe in this case since the resulting draws are even more dispersed. Figure 3 shows how the solution depends on $z^{\max}$. The effect of the truncation $z^{\max}$ will be model specific and in practice experimentation is required. If we, for example, simply set $z^{\max} = 1,000,000$, the variance of the simulated Bellman operator becomes very large for a given $N$ due to the issue with undefined sample weights $w_{N,i}(z, d)$ mentioned above. At the same time, choosing $z^{\max}$ too small leads to a large bias. To balance the bias and variance, we ended up using $z^{\max} = 1000$ which all subsequent numerical results for the self-approximating method is based on. Finally, we would like to stress that none of these issues appear for the sieve-based method.

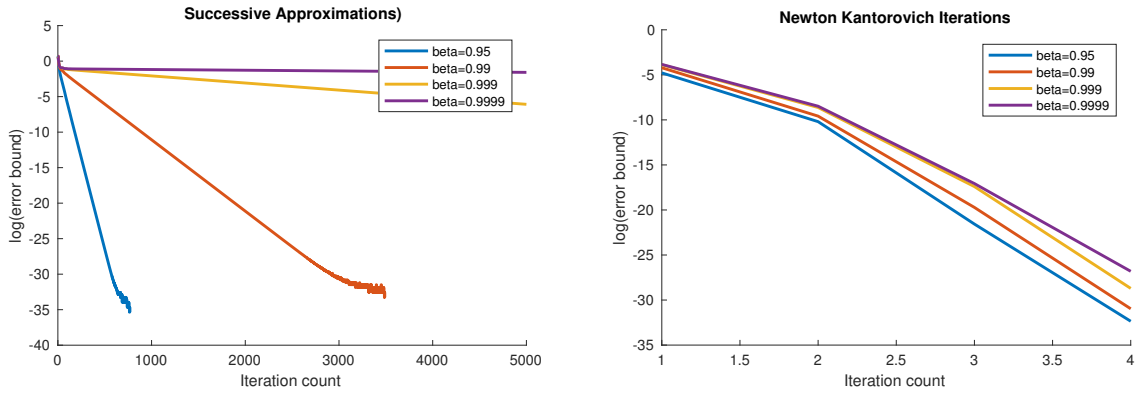
## 6.3 Convergence properties and computation times

We first investigate the convergence properties of our solution methods for given choice of $K$ and $N$. Do they converge and if so how fast?

### Global convergence properties of sieve method

As demonstrated in Theorem 1, the simulated Bellman operators are always contraction mappings and so the self-approximating method is guaranteed to converge using successive approximations. In contrast, $\Pi_K \bar{\Gamma}_{N,\lambda}$ is not necessarily a contraction and so global convergence of the

Figure 4: Convergence and discount factor




Notes: Discount factor is $\beta \in \{0.95, 0.99, 0.999, 0.9999\}$, utility function parameters are $\theta_c = 2$, $RC = 1$, $\lambda = 1$ and transition parameters are $\sigma_\varepsilon = 15$, $a = 2$, $b = 5$ and $\pi = 0.000000001$.

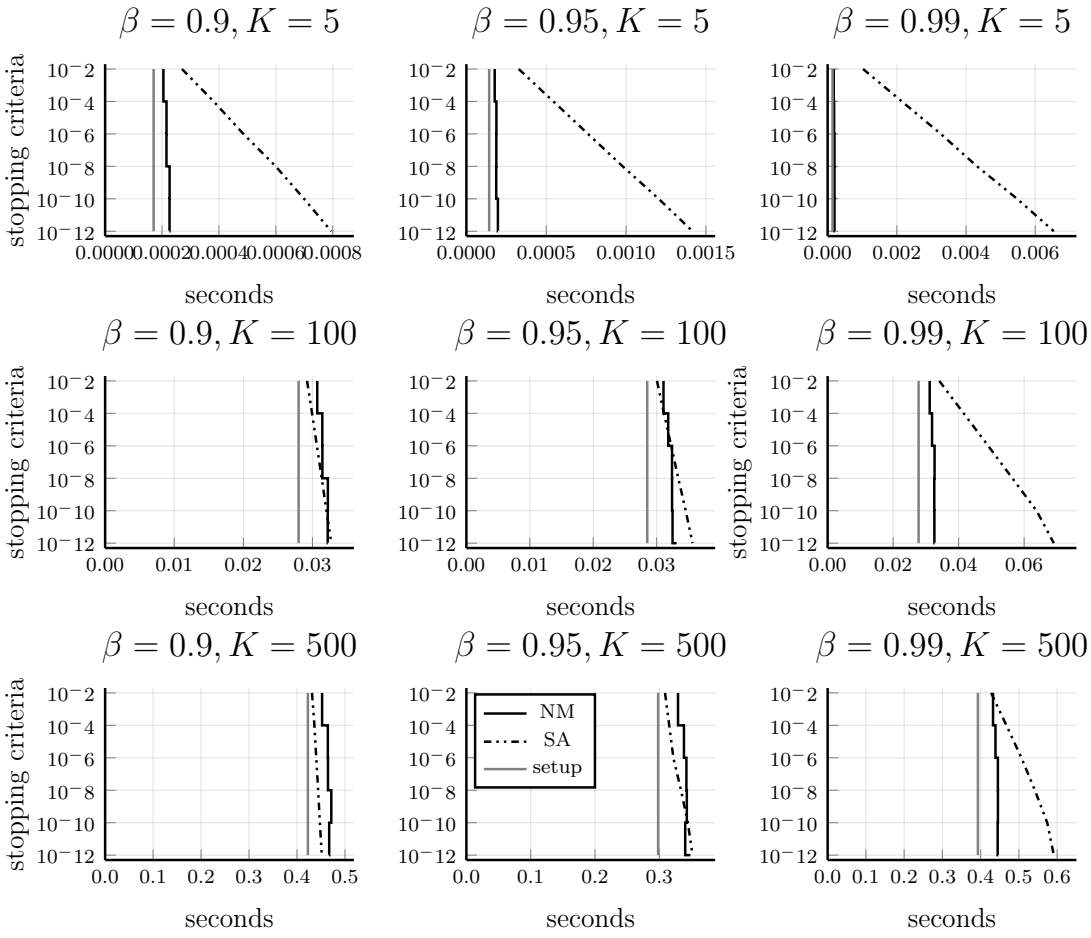
sieve method may fail, c.f. discussion in Section 5.2. A sufficient condition for global convergence is $||\Pi_K||_{op,\infty} < 1/\beta$ and we saw that $||\mathbf{P}_K||_{op,\infty} > 1$ implies $||\Pi_K||_{op,\infty} > 1$. However, even if $||\mathbf{P}_K||_{op,\infty} > 1$, successive approximation may still converge: Across various parameter values of model, choices of sieve spaces and number of simulations, we did not encounter any failure of the sieve method to converge and the resulting approximate solution was well-behaved. This finding held across various initializations of the solution algorithms (initial choice of sieve coefficients). For example, we implemented the sieve method using $M = 64$ evaluation points and using either $K = 1$ or $K = 4$ Chebyshev basis functions. We found that $||\mathbf{P}_1||_{op,\infty} = 1$ while $||\mathbf{P}_4||_{op,\infty} > 1.78$ and so the sieve method was guaranteed to converge for $K = 1$ but not for $K = 4$. Nevertheless, the method of successive approximations did in fact converge to a tolerance of $10^{-12}$ for both $K = 1$ and $K = 4$.

### Successive approximation versus Newton-Kantorovich

In Section 4.3 we advocated a hybrid of successive approximation (SA) and Newton-Kantorovich (NK) where we start with SA to ensure global convergence, and switch to NK iterations once the domain of attraction has been reached since NK generally converges faster. We illustrate this attractive feature of the NK algorighm in Figure 4 where we have plotted the log residual error of the current value function approximation (relative to the "exact" solution) against the iteration count for the SA and NK algorithms, respectively, for four different values of $\beta$. As expected, the convergence of the SA algorithm requires a very large number of iterations ($> 1000$) with computation time increasing in $\beta$, where as NK converges after less than 10 iterations and with the value of $\beta$ having little effect on its performance.

Figure 4 is silent about the over-all computation time of SA relative to NK. Compared to SA, each NK iteration is more expensive since the former only requires computing the simulated Bellman operator evaluated at the value function obtained in the previous step while the latter, in addition, requires computing its functional derivative and inverting a $K \times K$ dimensional

Figure 5: Run-times (incl setup times) for SA (dotted lines) and NK (drawn lines) algorithms.




matrix for the integrated value function and a $KD \times KD$ dimensional matrix for the expected value function, c.f. Section 4.3. With $K$ large, one could therefore fear that NK would become computationally too expensive.

In Figure 5 we report best of 10 run-times for various levels of $K$ and $\beta$ and tolerance levels of SA and NK where we also include set-up time (time spent on initial computations before starting the actual algorithm). As expected, we find that NK is the faster of the two algorithms when $\beta$ is relatively large and $K$ is relatively small. With $K = 5$ NK is faster across all levels of $\beta$ while for $K = 100$ and $K = 500$, SA is faster for moderate values of $\beta$. However, as we shall subsequently see, with $K = 5$ the sieve method carries almost no bias and so choosing $K$ larger (such as 100 or 500) is actually unnecessary here and is only included here to illustrate potential issues with NK for models where a large number of sieve terms are needed to obtain a good approximation of the value function. Moreover, in most empirical applications, $\beta$ is chosen to be larger than 0.99 in which case NK still dominates SA even with $K = 500$.

## 6.4 Approximation quality

We here investigate how the approximate value function is affected by the number of draws and the chosen projection basis. The goal is to demonstrate the rate results of the theoretical sections, and to compare the two types of basis functions spaces that we described above. We will take a partial approach and first fix $N$ to study the role of $K$, and then afterwards fix $K$ to study the role of $N$. All subsequent results are for the case of $\beta = 0.95$. This is to save space. We implemented the methods for other values of $\beta$ and since the numerical results were qualitatively the same, we have left these out. The main difference in the numerical results is that higher values of $\beta$ tend to shift the overall level of the value function upwards and add more curvature to it. This in turn generally leads to an increase in the absoute bias and variance numbers. However, in terms of percentage bias and variance, the performance of the methods were very similar across different values of $\beta$.

### Effect of varying $K$ for projection-based value function approximation

The theory for the projection-based value function approximation informs us that the choice of the basis functions will have a first-order effect on the bias while only a second-order effect on the variance. In particular, we expect $Bias(z)$, as defined in the beginning of this section, to satisfy $Bias(z) \cong \Pi_K(v)(z) - v(z)$, c.f. discussion following Theorem 5, while $Var(z)$ should be much less affected by $K$. The actual size of the bias obviously depends on the curvature and smoothness of $v_0$ and the particular choice of basis functions. But we know that $v$ is an analytic function and with only moderate curvature, c.f. Figure 1 and so expect it to be well-approximated by a small number of polynomial basis functions.

This is confirmed by the pointwise bias and standard deviation, $\sqrt{Var(z)}$, reported in Figure 6: First, as can be seen in the left-hand side panel of Figure 6, first-order B-splines lead to significantly larger point-wise bias compared to the other two sieve bases, namely second-order B-splines and Chebyshev polynomials. This is accordance with theory since we know that a smooth function is better approximated by higher-order polynomials, c.f. the error rates reported in Example 1 as a function of $s$. At the same time, second-order B-splines and Chebyshev polynomials exhibit very similar biases for a given choice of $K$.

The right-hand side panel of Figure 6 shows the point-wise standard deviation across different choices of $K$ for the three different sieve bases. Consistent with the theory, the standard deviation of the value function approximation is not very sensitive to the particular choice of the sieve basis and the number of basis functions uses. That is, the sieve basis mostly affect the bias with only minor impact on the variance.

Finally, we examine how the bias behaves as we further increase $K$. Figure 7 plots $\|Bias\|_\infty$ as a function of $K$. Similar to Figure 6 we see much more rapid convergence when smooth basis functions are used, and with little improvement for $K$ greater than 9. This is not surprising given the reported shape of $v$. The second-order B-splines and Chebyshev basis functions produce very similar fits, even if they are evaluated on different grids and the B-splines have very different

Figure 6: Point-wise bias and standard deviation of solutions for various choices of $K$ using different interpolation schemes, $N = 200$, $S = 200$, $\sigma_Z = 15$.

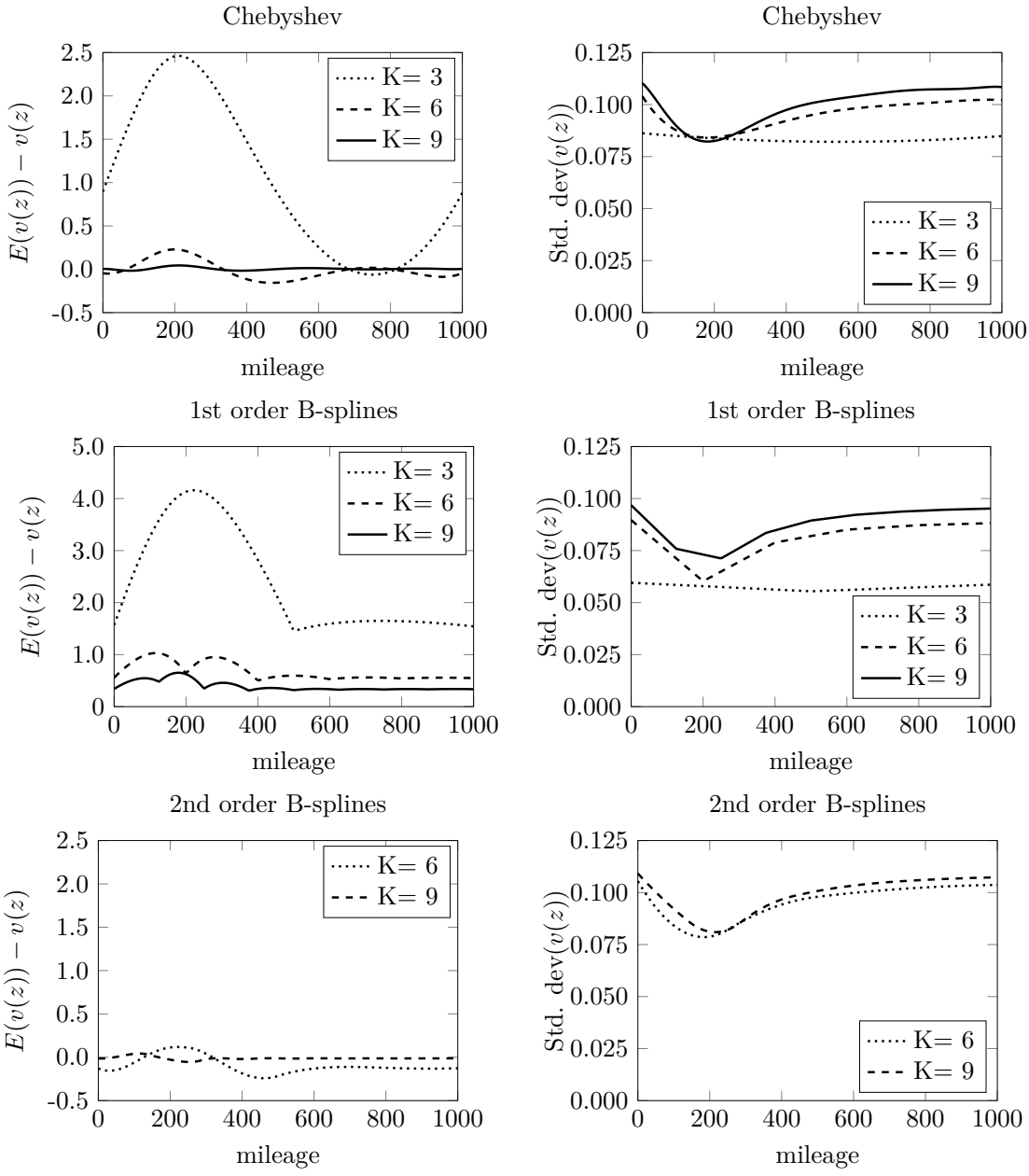
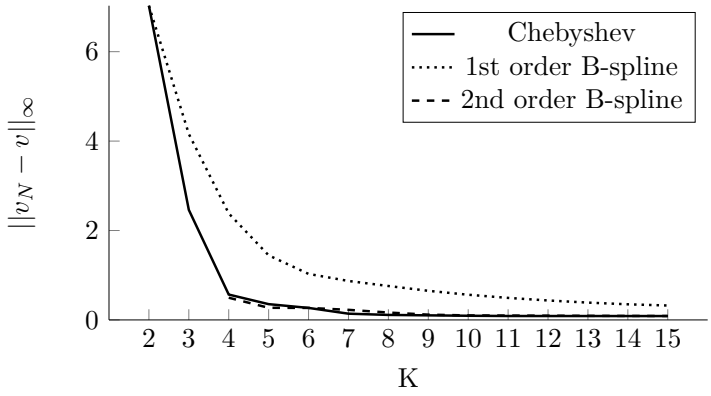


Figure 7: Sup-norm of bias of solutions for various choices of $K$ using various interpolation schemes, $N = 200$, $S = 200$.




properties compared to Chebyshev polynomials. Indeed, the curves are practically overlapping. This is in accordance with the asymptotic theory that predicts that higher-order B-splines and Chebyshev polynomials should lead to similar biases. Moreover, the theory informs us that if $v$ is analytic, and this is the case in this particular implementation, we should expect the bias to vanish with rate $O(K^{-K})$ when using polynomial interpolation. The bias indeed does go to zero very quickly and so the numerical results support the theory.

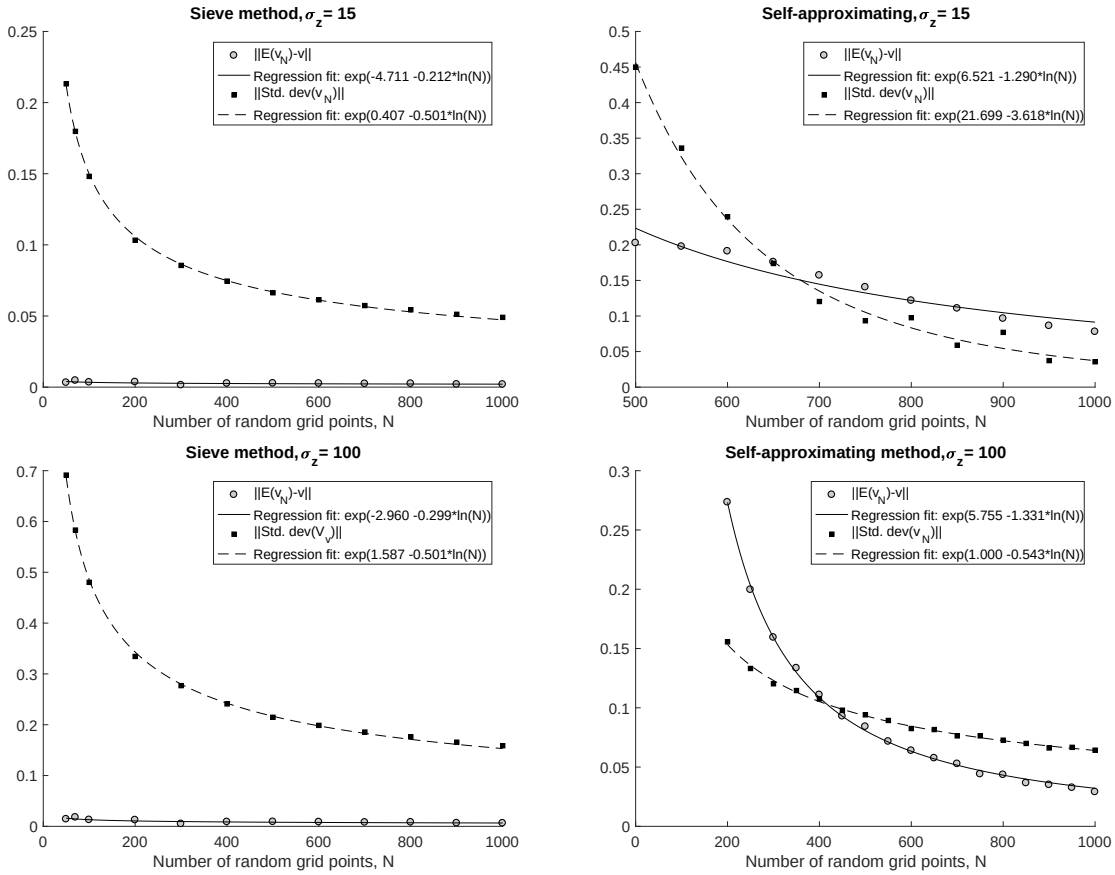
**Simulation errors, rates of convergence and asymptotic normality**

We now compare the errors due to simulations and the rates with which these vanish for the two solution methods. For both methods, theory tells us that $N$ should have a first-order effect on the variance of the approximate value function which is supposed to vanish at rate $1/N$, c.f. Theorems 5 and 3. Our asymptotic theory is, on the other hand, silent about the size of simulation bias and the rate with which it should vanish with. However, we can think of both the sieve-based and self-approximating method as a nonlinear GMM-estimator where the simulated Bellman operator defines the sample moments. Importing results for GMM estimators, see, e.g., Newey and Smith (2004), we should expect the simulation bias to be of order $1/N$.

In Figure 8 we investigate this prediction by plotting $\|Bias\|_\infty$ and $\left\|\sqrt{Var}\right\|_\infty$ for the sieve-based method (left panels) and for the self-approximating method (right panels) for two different choices of $\sigma_\varepsilon$ and for across different values of $N$. To examine the rate with which the simulation bias and variance vanish we estimate the following an exponential regressions by NLS $||\sqrt{Var}||_\infty = \exp(\alpha_{SD} + \rho_{SD} \ln(N))$ and $||Bias||_\infty = \exp(\alpha_{Bias} + \rho_{Bias} \ln(N))$ where $\rho_{SD}$ and $\rho_{Bias}$ measures the rate that $||\sqrt{Var}||_\infty$ and $||Bias||_\infty$ vanishes with respectively. The resulting regression fit estimates are reported in both Figure 8 as well as in Table 1. In Table 1 we present bias and standard deviation for $N = 500$ as well as their rates of convergence both methods; with various values of $K$ for the sieve approximation method.

According to the theory, the variance should vanish with rate $1/N$ for both methods and we

Figure 8: Convergence results




Notes: Discount factor is $\beta = 0.95$, utility function parameters are $\theta_c = 2$, $RC = 10$, and transition parameters are $a = 2$, $b = 5$ and $\pi = 0.000000001$. Uniform bias and variance were estimated using 500 evaluation points and $S = 2000$ implementations.

Table 1: Bias, variance, and rates of convergence for various values of K

| | Sieve Method | | | | | Self-approx. |
|---|---|---|---|---|---|---|
| # of basis functions, K | 1 | 2 | 5 | 10 | 15 | method |
| | | | $\sigma_z$=15 | | | |
| $\|\|Bias\|\|_\infty$ for $N = 500$ | 12.743 | 7.029 | 0.348 | 0.016 | 0.003 | 0.203 |
| $\|\|\sqrt{Var}\|\|_\infty$ for $N = 500$ | 0.000 | 0.020 | 0.063 | 0.066 | 0.066 | 0.450 |
| Convergence rate for $\|\|Bias\|\|_\infty$ | 0.000 | 0.000 | 0.002 | -0.012 | -0.212 | -1.290 |
| Convergence rate for $\|\|\sqrt{Var}\|\|_\infty$ | 0.169 | -0.500 | -0.501 | -0.501 | -0.501 | -3.618 |
| | | | $\sigma_z$=100 | | | |
| $\|\|Bias\|\|_\infty$ for $N = 500$ | 22.446 | 10.937 | 0.112 | 0.009 | 0.009 | 0.084 |
| $\|\|\sqrt{Var}\|\|_\infty$ for $N = 500$ | 0.000 | 0.128 | 0.218 | 0.215 | 0.215 | 0.094 |
| Convergence rate for $\|\|Bias\|\|_\infty$ | 0.000 | 0.000 | -0.027 | -0.299 | -0.299 | -1.331 |
| Convergence rate for $\|\|\sqrt{Var}\|\|_\infty$ | 0.169 | -0.500 | -0.501 | -0.501 | -0.501 | -0.543 |

therefore expect $\rho_{SD} = -0.5$ so that $||\sqrt{Var}||_\infty$ vanish with $1/\sqrt{N}$. For the projection based method, we see that the rate with which the standard deviation shrinks to zero is indeed close to $-0.5$ for all values of $K > 1$ and irrespectively of the value of $\sigma_\varepsilon$. For the self-approximating method we estimate the rate to $\rho_{SD} = -0.541$ when $\sigma_\varepsilon = 100$, which is in line with the theory. However, $||\sqrt{Var}||_\infty$ is found to vanish with rate $1/N^{3.6}$ for $\sigma_\varepsilon = 15$. This seems to indicate that the asymptotic theory developed in Theorems 3 and 4 do not provide a very accurate approximation of the performance of the self-approximating method for small and moderate choices of $N$ when the support of $Z_t|\,(Z_{t-1} = z, d_t = 1)$ is small ($\sigma_Z = 15$). We conjecture that the discrepancy between theoretical predictions and numerical results for the self-approximating method is due to the aforementioned issues with the marginal importance sampler discussed in Section 6.2: Many of the draws are not used in the computation of the simulated Bellman operator because they fall outside the support of $Z_t|Z_{t-1} = z$ for a given choice of $z$. Thus, the effective number of draws is smaller than $N$ and changes as $z$ varies.

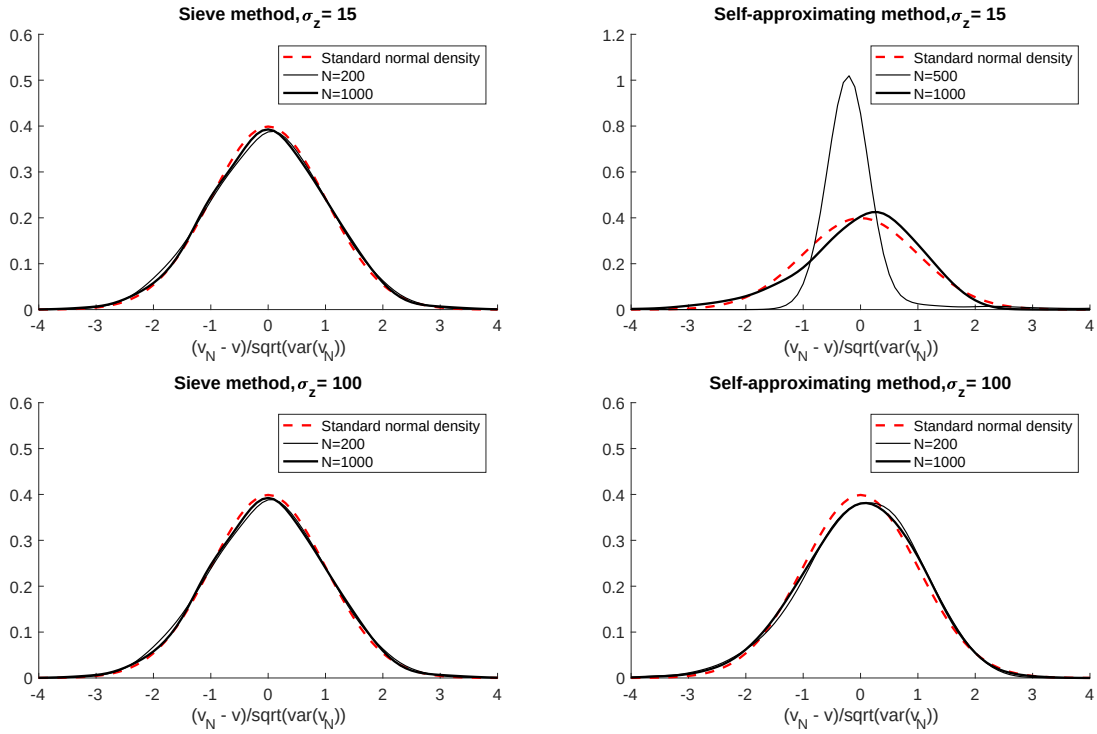
For the projection based method, the main source of bias is due to the sieve projection. From Figure 6, we see that, with $N = 200$ and $K = 9$, the sieve-based methods using second-order B-splines or Chebyshev polynomials have virtually no bias, and both Figure 8 as well as in Table 1 also confirms that we practically eliminate by approximate the value function using Chebychev polynomials with $K = 20$. However, there still remains a small bias that vanishes as $N$ grows. For small $K$, we see that the bias is roughly independent of $N$. As $K$ increases so does the dependence on $N$. However, even for $K = 20$ where we estimate $\rho_{Bias}$ to be 0.21 and 0.30 for $\sigma_Z = 15$ and $\sigma_Z = 100$ respectively, the rate of convergence is far from $1/N$. This is probably due to the presence of higher-order bias components that our asymptotic theory does not account for.

For the self-approximating method, there is no sieve projection bias but a larger simulation induced bias that decreases with $N$. We obtain rate estimates of $1/N^{1.7}$ and $1/N^{1.4}$ for the bias when $\sigma_Z = 15$ and $\sigma_Z = 100$ respectively; these are slightly faster than expected but not too

far from the theoretical predictions of $1/N$. For the self-approximating method, bias constitute more than half of RMSE when $N < 600$ for $\sigma_Z = 15$ (or $N < 400$ for $\sigma_Z = 100$), but since $||Bias||_\infty$ decays faster than $||\sqrt{Var}||_\infty$ , the simulation bias eventually becomes second order for large $N$.

Comparing $||MSE||_\infty$ for $N = 500$ we find that the sieve-based method clearly dominates the self-approximating method when $\sigma_Z = 15$, whereas the self-approximating method performs best when $\sigma_Z = 100$. This is not entirely surprising since a large value of $\sigma_\varepsilon$ implies a large conditional support of $Z_t$ in which case the draws of the marginal sampler are more likely to fall within the support, c.f. the discussion in Subsection 6.2. Thus, the over-all error of the self-approximating method will tend to be smaller when $\sigma_Z$ is large. The opposite is the case for the sieve based method which becomes more precise for smaller value of $\sigma_Z$ since the variance of the simulated Bellman operator used for this method gets smaller as $\sigma_Z$ gets smaller. This shows that there is considerably scope for improving the performance of the sieve-based method by more careful design of the sampling method.

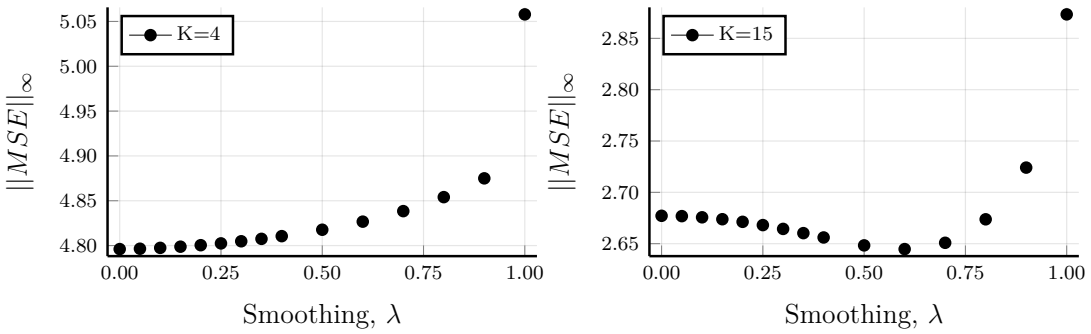
Figure 9: Asymptotic Normality




Notes: Each panel shows kernel density estimates of $(\widehat{v}_N(z) - E[\widehat{v}_N(z)])/\sqrt{var(\hat{v}_N(z))}$ for $z = 500$ based on $S = 2000$ solutions for each sample size $N$. Discount factor is $\beta = 0.95$, utility function parameters are $\theta_c = 2$, $RC = 10$, and transition parameters are $\sigma_Z = 100$, $a = 2$, $b = 5$ and $\pi = 0.000000001$.

Theorems 4 and 5 state that when $N$ is large, the approximate value functions should be normally distributed. We here investigate whether this asymptotic approximation is useful in practice by looking at the pointwise distribution of the approximate solutions obtained through both methods. In Figure 9, we plot the distribution of $(\tilde{v}(z) - E[\tilde{v}(z)])/\sqrt{Var(\tilde{v}(z))}$

Figure 10: Sup-norm MSE of solutions to Bellman operators with simulated taste shocks and state transitions for varying levels of smoothing, for $N = 100$.




for $z = 500$, where $\tilde{v}$ denotes a given approximation method, together with the standard normal distribution. It is here important to note we do not center the estimate around $v(z)$ but instead around $E[\tilde{v}(z)]$ ; this is due to the sizable bias of the self-approximating method. For the sieve-based method, we see that its normalized distribution is quite close to the standard normal irrespectively of the value of $\sigma_Z$. In contrast, the normal distribution is a poor approximation for the self-approximating method when $\sigma_Z = 15$ when $N = 500$; we expect this is due to the fact that the effective number of draws is quite small and so the asymptotic approximation is poor in this case. As expected the approximation gets better as $N$ and/or $\sigma_Z$ increases.

## 6.5 Effect of smoothing

The results reported above did not involve any smoothing bias. We now numerically study the effect of smoothing. This is done by, instead of integrating out the i.i.d. extreme value taste shocks $\varepsilon_t$ analytically as we have done so far, using Monte Carlo simulations to evaluate this part of the integral and then introducing the smoothing device to ensure that the simulated Bellman operator remains smooth. While this may appear somewhat artificial, the merit of doing this exercise is that we can use the same "exact" solution as benchmark as used above.

In Figure 10 we plot the sup-norm of the mean squared error, $||\text{MSE}||_\infty$ as a function of $\lambda$ (the smoothing scale parameter) for the sieve-based method using $K = 4$ or 8 Chebyshev polynomials (similar results were obtained for the self-approximating method and so are left out). For $K = 4$, the MSE increases monotonically as a function of $\lambda$ while for $K = 15$ the bias due to smoothing is non-monotonic in $\lambda$. In both cases, at $\lambda = 0$, any remaining biases are due to either sieve-approximation or simulations. Importantly, the bias due to smoothing is negiglible (relative to the other biases) for small and moderate values of $\lambda$ while the variance is largely unaffected. We have no theory or heuristics for choosing an optimal $\lambda$ to optimally balance bias and variance due to smoothing but the current numerical results indicate that choosing a quite small $\lambda$ value works well.

## 6.6 Performance in the bivariate case

We now examine how the solution methods perform in the bivariate case ($d_Z = \dim(Z_t) = 2$) in order to see if there is any curse of dimensionality built into the two methods. We do this for two different models as described below.

### An Additive DDP

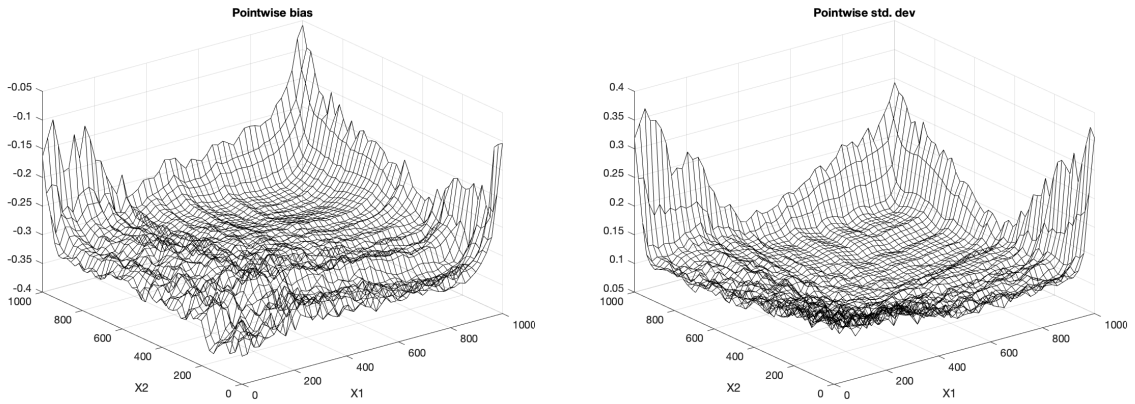
We here follow the approach of Arcidiacono et al. (2013) and Rust (1997a) and build a $d_Z$-dimensional model by adding up $d_Z$ independent versions of the univariate model considered so far. That is, we choose the utilities and state dynamics as $\bar{u}(z, \varepsilon, d) = \sum_{i=1}^{d_Z} u(z_i, \varepsilon_i, d_i)$ and $\bar{F}_Z(z'|z, d) = \prod_{i=1}^{d_Z} F_Z(z_i'|z_i, d_i)$, where $z = (z_1, ..., z_{d_z})$ and $d = (d_1, ..., d_{d_Z})$, with $F_Z(z_i'|z_i, d_i)$ and $u(z_i, \varepsilon_i, d_i)$ denoting the state transition and per-period utility in the univariate case as described in Section 6.1. Note here that $(Z_{t,i}, \varepsilon_{t,i})$ and $(Z_{t,j}, \varepsilon_{t,j})$ are fully independent of each other, $i \neq j$ and the number of alternatives are $2^{d_Z}$, where $d_Z$ Thus, the model considers the joint replacement decision of $d_Z$ assets whose stochastic usages $(Z_{t,1}, ..., Z_{t,d_Z})$ are mutually independent. Conveniently, the integrated value function of this multidimensional problem, $\bar{v}(z_1, ..., z_{d_Z})$, is simply the sum of the solutions to each of the underlying univariate models, $\bar{v}(z_1, ..., z_{d_Z}) = \sum_{i=1}^{d_Z} v(z_i)$, where $v(z_i)$ is the solution to the univariate model in Section 6.1. This is a rather simplistic multivariate model but it comes with the major advantage that we can obtain a very accurate approximation of the exact solution by simply adding up the "exact" solution found for the univariate case. With a more complicated multidimensional structure, the computational cost of finding the "exact" solution is much higher. However, when implementing our solution methods, we forgo forgo the knowledge of the additive structure of the solution and so treat the above model as a "proper" multivariate problem.

### Simulation error

Given the issues with the self-approximating method for small values of $\sigma_Z = 15$, we here focus exclusively on the case $\sigma_Z = 100$. To get a sense of the pointwise performance of the self-approximating method, we plot the pointwise bias and standard deviation for this method with $N = 3000$ in Figure 11 together with the pointwise errors of the corresponding replacement (choice) probabilities. The overall shape and level of the integrated value function is quite well captured, and the same is true for the policy. However, the approximation errors tend to get larger out in the tails of the distribution and some of this comes from the fact that the issues with the marginal sampler used for the self-approximating method are amplified here. The problems are especially present in the off-grid evaluations, where we often have very few draws in a given region where we want to evaluate the value function or policies.

Next, we examine $||Bias||_\infty$ and $||\sqrt{Var}||_\infty$ for both methods as we increase $N$. These are plotted in Figure 12 where it should be noted that the reported range of $N$ reported on the $x$-axis of the two figures differ substantially. This is due to the fact that the self–approximating method became numerically unstable for $N$ smaller than 1,400 while no such issues were present

Figure 11: Approximation errors of self-approximating method, bivariate DDP




Notes: Discount factor is $\beta = 0.95$, utility function parameters are $\theta_c = 2$, $RC = 10$, $\lambda = 1$ and transition parameters are $\sigma_Z = 100$, $a = 2$, $b = 5$ and $\pi = 0.000000001$. The "exact" solution was computed by averaging over $S = 100$ solutions, each found using the smoothed random Bellman operator with $N = 3000$ pseudo random draws. Each fixed point was found using a contraction tolerance of machine precision.

for the sieve-based method. As in the univariate case, both bias and variance of the two methods vanish as $N$ increases. However, comparing Figures 12 and 8, while the errors of the sieve-based method in the bivariate case is of a similar magnitude as in the univariate case, the errors of the self-approximating method are much larger in the bivariate case. This seems to indicate a certain type of curse-of-dimensionality in this particular application of the self-approximating method. This is caused by the issues with the marginal importance sampler employed for this method.

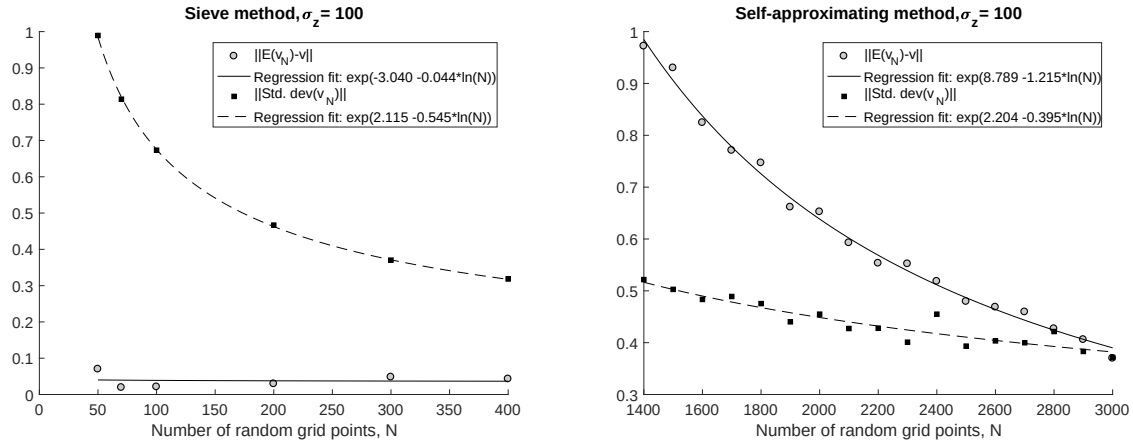
### Sieve approximation error

In the implementation of the sieve-based method we use as sieve basis the tensor product of univariate Chebyshev polynomials or B-splines. That is, given, say, $J$ univariate basis functions, say, $p_1, ..., p_J$, we construct our bivariate basis functions as $B_{i,j}(z_1, z_2) = p_i(z_1) p_j(z_2)$ for $i, j = 1, ..., J$ yielding a total of $K = J^2$ bivariate basis functions. In particularly, we do not exploit the additive structure of the problem since we are interested in the practical contents of Theorems 5 where no particular sparsity/special structure of the model is assumed to be known.

However, in practive, Chebyshev polynomials very easily pick up the additive structure and effectively sets the coefficients of the cross-product terms to zero. This is illustrated in Table 2 in Appendix C, where we report the coefficients for one particular projection-based bivariate value function estimate using a tensor product of $J = 5$ Chebyshev polynomials. However, this is due to the particular properties of the Chebyshev polynomials and is not enforced by us in the implementation. For example, if we instead use B-splines, the "estimated" coefficients of the cross-product terms were significantly different from zero, c.f. Table 3 in Appendix C.

In the left-hand side panel (a) of Figure 13, we report the uniform bias of the projection-

Figure 12: Simulation errors for bivariate additive DDP




Notes: Discount factor is $\beta = 0.95$, utility function parameters are $\theta_c = 2$, $RC = 10$, $\lambda = 1$ and parameters for transition density $f(z'|z,d)$ are $\sigma_Z = 100$, $a = 2$, $b = 5$ and $\pi = 0.000000001$. Point-wise bias and variance was estimated in 500 evaluation points based on $S = 200$ replications. We report the sup norm of the bias and the standard deviation for each N for both methods and NLS regression fits of $||\sqrt{Var}\}||_\infty = \exp(\alpha_{SD} + \rho_{SD} \ln(N)$ and $||Bias||_\infty = \exp(\alpha_{Bias} + \rho_{Bias} \ln(N)$.

based method with $N$ chosen very large for the additive bivariate model. We find that the bias vanishes as $K$ increases as in the one-dimensional model. However, convergence is now slower in $K$ relative to the univariate case and we require $K = 50$ to obtain a sieve approximation bias of $10^{-2}$ while $K = 7$ sufficed in the univariate case. This is consistent with theoretical error rates for polynomial interpolation where the rate slows down as the dimension of the problem increases, c.f. Section 4.3.

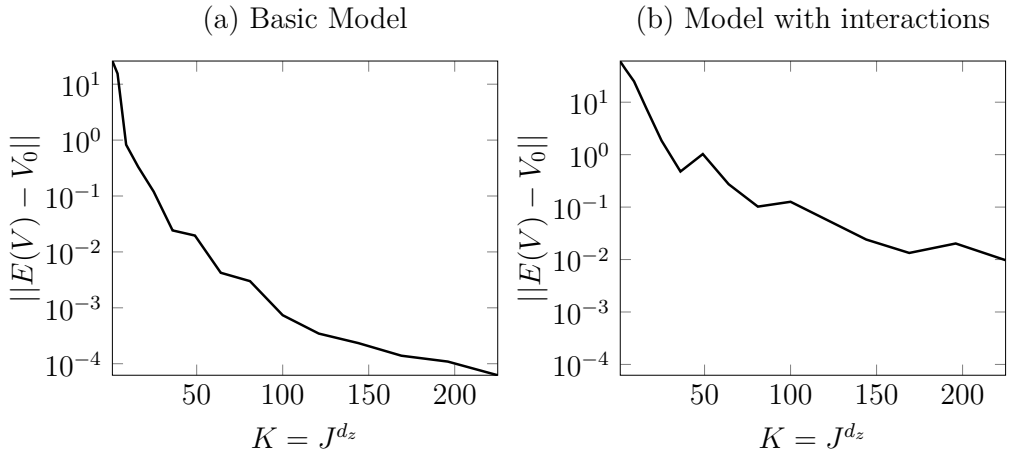
### A non-additive DDP

One concern with the numerical results reported for the bivariate additive model in the previous section is that they may understate the curse of dimensionality of the sieve method: The true value function is by construction additive in the two state variables and so interaction terms do not appear. This in turn implies that the computational complexity of solving this particular model is relatively low; in particular, the solution should be well-approximated by lower-dimensional sieves ($K$ small).

To investigate how the sieve method performs when applied to a more complex, non-additive model, we here consider a slightly more complicated bivariate model where we include a multiplicative interaction term so that maintenance and replacement costs of the two busses interact, $\bar{u}(z,d) = \sum_{i=1}^{2} u(z_i, d_i) - u(z_1, d_1)u(z_2, d_2)/20$. Such a structure could, for example, reflect that capacity constraints make it more costly to simultaneously replace the engines of both busses. The resulting value funtion will have a more complicated multidimensional structure and so we expect that the computational cost of our sieve method should be higher in this scenario.

The sieve approximation bias of our solution method for this model is reported in the right-hand side panel (b) in Figure 13. Compared to panel (a) – the additive case – we see that

Figure 13: Bias of value function in bivariate DDP for varying $K$ .




more sieve terms are required in order to reach a specific absolute error level in the model with interactions. In Table 4 in Appendix C the coefficients on the first ten basis functions in each dimension and their interactions are reported. Compared to the Chebyshev-based solution earlier we see quite significant coefficients on the coefficients for the cross-terms. However, the coefficients on the basis functions tend to zero quite quickly as K increase. The sup-norm of the difference in the value function at 40.000 evaluation grids is on the order of $10^{-5}$ when comparing the solutions with $K = 50^2 = 2500$ and $K = 30^2 = 900$ basis functions, and individual coefficients fall below $10^{-6}$ for univariate basis functions and cross products beyond the 22nd univariate basis functions, and below $10^{-8}$ around the 30th basis functions.However, it is important to stress that a large $K$ here only comes with a computational cost while a large $K$ has little effect on the variance of the sieve method. All together, we find that the sieve-based solution method works well also in higher dimensions, in particular when the model has a particular structure that can be utilized in the solution method.

# 7 Conclusion

We have proposed two novel methods for numerical computation of either the so-called integrated or expected value functions in a general class of dynamic discrete choice models.. Both methods rely on a smoothed simulated version of the Bellman operators definining the integrated and expected values functions. The smoothing facilitates both the practical implementation and the theoretical analysis of the approximate value functions. Under regularity conditions, we develop an asymptotic theory for the two methods as the number of simulations used to compute the simulated Bellman operators diverge. A set of numerical experiments show that our first method, the so-called self-approximating method can be somewhat unstable while the second one, which relies on sieve methods, apppears much more numerically robust. The next step is to develop methods for choosing the number of simulations, sieve basis functions

and smoothing parameter $\lambda$ in a given setting so that the resulting approximate solution is of a good quality. Another area of research is to investigate how the proposed solution methods can be used for the estimation of dynamic discrete choice models.

# References

Arcidiacono, P., P. Bayer, F. A. Bugni, and J. James (2013). Approximating high-dimensional dynamic models: Sieve value function iteration. *Advances in Econometrics 31*, 45–95.

Bowman, A., P. Hall, and T. Prvan (1998). Bandwidth selection for the smoothing of distribution functions. *Biometrika 85*, 799–808.

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

Cai, Y. and K. L. Judd (2013). Shape-preserving dynamic programming. *Mathematical Methods of Operations Research 77*(3), 407–421.

Chen, V. C. (1999). Application of orthogonal arrays and mars to inventory forecasting stochastic dynamic programs. *Computational Statistics & Data Analysis 30*, 317–341.

Chen, X. (2007). Large sample sieve estimation of semi-nonparametric models. In J. J. Heckman and E. E. Leamer (Eds.), *Handbook of Econometrics*, Volume 6. Elsevier.

Chen, X. and H. White (1999). Improved rates and asymptotic normality for nonparametric neural network estimators. *IEEE Transactions on Information Theory 45*(2), 682–691.

Fermanian, J.-D. and B. Salanie (2004). A nonparametric simulated maximum likelihood estimation method. *Econometric Theory 20*, 701–734.

Iskhakov, F., T. H. Jørgensen, J. Rust, and B. Schjerning (2017). The endogenous grid method for discrete-continuous dynamic choice models with (or without) taste shocks. *Quantitative Economics 8*(2), 317–365.

Judd, K. L., L. Maliar, S. Maliar, and R. Valero (2014). Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. *Journal of Economic Dynamics and Control 44*, 92–123.

Keane, M. and K. I. Wolpin (1994). The solution and estimation of discrete choice dynamic programming models by simulation and interpolation: Monte carlo evidence. *The Review of Economics and Statistics 76*, 648–672.

Kristensen, D. and B. Salanie (2017). Higher order properties of approximate estimators. *Journal of Econometrics 198*, 189–208.

Kristensen, D. and Y. Shin (2012). Estimation of dynamic models with nonparametric simulated maximum likelihood. *Journal of Econometrics 167*, 76–94.

Lizotte, D. J. (2011). Convergent fitted value iteration with linear function approximation. In *NIPS'11 Proceedings of the 24th International Conference on Neural Information Processing Systems*, pp. 2537–2545.

Lumsdaine, R., J. Stock, and D. Wise (1992). Three models of retirement: Computational complexity versus predictive validity. In D. Wise (Ed.), *Topics in the Economics of Aging*, pp. 21–60. NBER: University of Chicago Press.

Ma, Q. and J. Stachurski (2020). Dynamic programming deconstructed: Transformations of the bellman equation and computational efficiency. *Operations Research forthcoming*.

McFadden, D. (1989). A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica 57*, 995–1026.

Munos, R. and C. Szepesvari (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research 1*, 815–857.

Newey, W. K. and R. J. Smith (2004). Higher order properties of gmm and generalized empirical likelihood estimators. *Econometrica 72*, 219–255.

Norets, A. (2010). Continuity and differentiability of expected value functions in dynamic discrete choice models. *Quantitative economics 1*, 305–322.

Norets, A. (2012). Estimation of dynamic discrete choice models using artificial neural network approximations. *Econometric Reviews 31*, 84–106.

Pal, J. and J. Stachurski (2013). Fitted value function iteration with probability one contractions. *Journal of Economic Dynamics & Control 37*, 251–264.

Rivlin, T. J. (1990). *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. New York: Wiley-Interscience.

Robert, C. and G. Casella (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.

Rust, J. (1987). Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica 55*, 999–1033.

Rust, J. (1988). Maximum likelihood estimation of discrete control processes. *SIAM Journal on Control and Optimization 26*(5), 1006–1024.

Rust, J. (1997a). A comparison of policy iteration methods for solving continuous-state, infinite-horizon markovian decision problems using random, quasi-random, and deterministic discretizations.

Rust, J. (1997b). Using randomization to break the curse of dimensionality. *Econometrica 65*, 487–516.

Rust, J. (2008). Dynamic programming. *The New Palgrave Dictionary of Economics: Volume 1–8*, 1471–1489.

Rust, J., J. Traub, and H. Wozniakowski (2002). Is there a curse of dimensionality for contraction fixed points in the worst case? *Econometrica 70*, 285–329.

Schumaker, L. L. (2007). *Spline Functions: Basic Theory, Third Edition*. Cambridge University Press.

Tjahjowidodo, T. et al. (2017). A direct method to solve optimal knots of b-spline curves: An application for non-uniform b-spline curves fitting. *PloS one 12*(3), e0173857.

van der Vaart, A. and J. Wellner (2000). Preservation theorems for glivenko-cantelli and uniform glivenko-cantelli classes. In W. J. Gine E., Mason D.M. (Ed.), *High Dimensional Probability II. Progress in Probability*, Volume 47.

van der Vaart, A. W. and J. A. Wellner (1996). *Weak Convergence and Empirical Processes*. Springer.

# A Auxiliary Results

We derive two general results for approximate solutions to functional fixed-points. Let $(\mathcal{X}, \|\cdot\|)$ be a normed vector space and $\Psi : \mathcal{X} \to \mathcal{X}$ be some contraction mapping w.r.t. $\|\cdot\|$ so that there exists a unique solution $x_0 \in \mathcal{X}$ to $x = \Psi(x)$. Let $\Psi_N$ be an approximation to $\Psi$ and let $\Pi_K$ be a projection operator, $K, N \geq 1$.

**Theorem A.1.** *Suppose (i) $\|\Psi_N(x_0) - \Psi(x_0)\| = O_p(\rho_{\Psi,N})$ for some $\rho_{\Psi,N} \to 0$ and (ii) for some $\beta < 1$, $\|\Psi_N(x) - \Psi_N(y)\| \leq \beta\|x - y\|$ for all $N$ large enough and all $x, y$. Then there exists a unique solution $x_N \in \mathcal{X}$ to $x = \Psi_N(x)$ with probability approaching one (w.p.a.1) satisfying $\|x_N - x_0\| = O_P(\rho_{\Psi,N})$.*

*Suppose furthermore (iii) $\Pi_K : \mathcal{X} \to \mathcal{X}$ satisfies $\|\Pi_K(x_N) - x_N\| = O_p(\rho_{\Pi,K})$ for some $\rho_{\Pi,K} \to 0$. Then there exists a unique solution $\hat{x}_N \in \mathcal{X}$ to $x = (\Pi_K \Psi_N)(x)$ w.p.a.1 satisfying*

$$\|\hat{x}_N - x_0\| \leq \|\hat{x}_N - x_N\| + \|x_N - x_0\| = O_p(\rho_{\Pi,K}) + O_p(\rho_{\Psi,N}).$$

*Proof.* We first observe that due to (ii), there exists a unique solution $x_N = \Psi_N(x_N)$ for all $N$ large enough which satisfies

$$\begin{aligned}\|x_N - x_0\| = \|\Psi_N(x_N) - \Psi(x_0)\| &\leq \|\Psi_N(x_N) - \Psi_N(x_0)\| + \|\Psi_N(x_0) - \Psi(x_0)\| \\ &\leq \beta\|x_N - x_0\| + \|\Psi_N(x_0) - \Psi(x_0)\|,\end{aligned}$$

and so $\|x_N - x_0\| \leq \|\Psi_N(x_0) - \Psi(x_0)\| / (1 - \beta) = O_P(\rho_{\Psi,N})$. Next, combining (ii) and (iii), we see that $\Pi_K \Psi_N$ is a contraction mapping w.p.a.1. with Lipschitz coeffient $\beta$, and so $\hat{x}_N$ defined in the theorem exists and is unique w.p.a.1. Moreover, by the same arguments employed in the analysis of $x_N$,

$$\|\hat{x}_N - x_N\| \leq \frac{\|\Pi_K \Psi_N(x_N) - \Psi_N(x_N)\|}{1 - \beta} = \frac{\|\Pi_K(x_N) - x_N\|}{1 - \beta} = O_p(\rho_{\Pi,K}).$$

□

**Theorem A.2.** *Suppose the following conditions are satisfied: (i) $\|x_N - x_0\| = O_P(\rho_{\Psi,N})$; (ii) $\rho_{\Psi,N}^{-1}\{\Psi_N(x_0) - \Psi(x_0)\} \rightsquigarrow \mathbb{G}$ in $(\mathcal{X}, \|\cdot\|)$; (iii) $\Psi_N(x_0)$ is Frechet differentiable at $x_0$ w.p.a.1 with Frechet differential $\nabla\Psi_N(x_0)[\cdot] : \partial\mathcal{X} \mapsto \mathcal{X}$ for some function set $\partial\mathcal{X}$, where $x_N - x_0 \in \partial\mathcal{X}$ w.p.a.1, such that $\|\Psi_N(x_N) - \Psi_N(x_0) - \nabla\Psi_N(x_0)[x_N - x_0]\| = o_P(\|x_N - x_0\|)$ and (iv) $\sup_{dx \in \partial\mathcal{X}: \|dx\| = 1} \|\{\nabla\Psi_N(x_0) - \nabla\Psi(x_0)\}[dx]\| = o_p(1)$. Then $\{I - \nabla\Psi(x_0)\}[\rho_{\Psi,N}\{x_N - x_0\}] \rightsquigarrow \mathbb{G}$. If furthermore (v) $I - \nabla\Psi(x_0)[\cdot] : \partial\mathcal{X} \mapsto \mathcal{X}$ has a continuous inverse, then $\rho_{\Psi,N}\{x_N - x_0\} \rightsquigarrow \{I - \nabla\Psi(x_0)\}^{-1}[\mathbb{G}]$.*

*Proof.* To show the first claim, combine a functional Taylor expansion with conditions (i) and

(iii),

$$\begin{aligned} 0 &= (I-\Psi_N)(x_N) = (I-\Psi_N)(x_0) + \{I-\nabla\Psi_N(x_0)\}[x_N-x_0] + o_P(\|x_N-x_0\|) \\ &= \Psi(x_0) - \Psi_N(x_0) + \{I-\nabla\Psi_N(x_0)\}[x_N-x_0] + o_P(\rho_{\Psi,N}). \end{aligned}$$

Next, by (iv),

$$\begin{aligned} & \|\{I-\nabla\Psi_N(x_0)\}[x_N-x_0] - \{I-\nabla\Psi(x_0)\}[x_N-x_0]\| \\ = \; & \|\{\nabla\Psi_N(x_0)-\nabla\Psi(x_0)\}[x_N-x_0]\| \\ \leq \; & \sup_{dx\in\partial\mathcal{X}:\|dx\|=1} \|\{\nabla\Psi_N(x_0)-\nabla\Psi(x_0)\}[dx]\| \, \|x_N-x_0\| \\ = \; & o_P(\rho_{\Psi,N}). \end{aligned}$$

Combining this with (ii),

$$\{I-\nabla\Psi(x_0)\}\left[\rho_{\Psi,N}^{-1}\{x_N-x_0\}\right] = \rho_{\Psi,N}^{-1}\{\Psi_N(x_0)-\Psi(x_0)\} + o_P(1) \rightsquigarrow \mathbb{G},$$

The second claim follows by (v) and the continuous mapping theorem. □

It is important here to note the tension between the requirement that that $x_N - x_0 \in \partial\mathcal{X}$ in (iii) and$\sup_{dx\in\partial\mathcal{X}:\|dx\|=1} \|\{\nabla\Psi_N(x_0)-\nabla\Psi(x_0)\}[dx]\| = o_p(1)$. The first condition will hold if we choose $\partial\mathcal{X}$ large enough. But at the same time, we need to show uniform convergence over the same space which will generally only hold if $\partial\mathcal{X}$ is Glivenko-Cantelli. In the application to value function approximation, this is achieved by choosing $\partial\mathcal{X} = \mathbb{C}_r^1(\mathcal{Z})$ defined in Section 4.3.3 for some $r < \infty$.

# B Proofs

*Proof of Theorem 1.* First note that for any $V(z;\lambda) \in \mathbb{B}\left(\mathcal{Z}\times\left(0,\bar{\lambda}\right)\right)^D$ and with $C$ denoting a generic constant,

$$\begin{aligned} |G_\lambda(u_\psi(U,z) + \beta V_\psi(U;z,\lambda))| &\leq C(1 + \|u_\psi(U,z)\| + \beta\|V_\psi(U;z,\lambda)\|) \\ &\leq C(1+\bar{u}_\psi(U) + \beta\|V\|_\infty), \end{aligned} \tag{B.1}$$

and so, using Assumption 1,

$$\begin{aligned} \|\Gamma(V)\|_\infty &\leq \sup_{(z,\lambda)\in\mathcal{Z}\times\left(0,\bar{\lambda}\right)} E\left[|G_\lambda(u_\psi(U;z) + \beta V_\psi(U;z,\lambda))| \, w_\psi(U;z)\right] \\ &\leq CE\left[|(1+\bar{u}_\psi(U) + \beta\|V\|_\infty)| \, \bar{w}_\psi(U)\right] < \infty, \end{aligned}$$

which shows that $\Gamma : \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D \mapsto \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$. Recycling eq. (B.1),

$$\begin{aligned}
\left\|\Gamma_N(V)\right\|_\infty &\leq \frac{\sum_{i=1}^N \left\|G_\lambda\left(u_\psi\left(U_i;z\right)+\beta V_\psi\left(U_i;z,\lambda\right)\right)\right\| w_\psi\left(U_i;z\right)}{\sum_{i=1}^N w_\psi\left(U_i;z\right)} \\
&\leq C\left(\frac{\sum_{i=1}^N \left\|\bar{u}_\psi\left(U\right)\right\| w_\psi\left(U_i;z\right)}{\sum_{i=1}^N w_\psi\left(U_i;z\right)} + 1 + \beta\left\|V\right\|_\infty\right) \\
&\leq C\left(\frac{\sum_{i=1}^N \left\|\bar{u}_\psi\left(U_i\right)\right\| \bar{w}_\psi\left(U_i\right)}{\inf_{z\in\mathcal{Z}}\sum_{i=1}^N w_\psi\left(U_i;z\right)} + 1 + \beta\left\|V\right\|_\infty\right) < \infty.
\end{aligned}$$

Thus, for any given $N \geq 1$, $\Gamma_N : \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D \mapsto \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$. To show that $\Gamma_N : \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D \mapsto \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ is a contraction, use that, by quasi-linearity of $G_\lambda(r)$, for any $V_1, V_2 \in \mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$,

$$\begin{aligned}
\Gamma_N(V_1)(z,\lambda,d) &= \sum_{i=1}^N G_\lambda\left(u_\psi\left(U_i;z\right)+\beta V_{\psi,2}\left(U_i;z,\lambda\right)+\beta\left[V_{\psi,1}\left(U_i;z,\lambda\right)-V_{\psi,2}\left(U_i;z\right)\right]\right) w_{N,i}\left(z,d\right) \\
&\leq \sum_{i=1}^N G_\lambda\left(u_\psi\left(U_i;z\right)+\beta V_{\psi,2}\left(U_i;z,\lambda\right)+\beta\left\|V_1-V_2\right\|_\infty \mathbf{1}_D\right) w_{N,i}\left(z,d\right) \\
&= \sum_{i=1}^N G_\lambda\left(u_\psi\left(U_i;z\right)+\beta V_{\psi,2}\left(U_i;z,\lambda\right)\right) w_{N,i}\left(z,d\right) + \beta\left\|V_1-V_2\right\|_\infty \sum_{i=1}^N w_{N,i}\left(z,d\right) \\
&= \Gamma_N(V_2)(z,\lambda,d) + \beta\left\|V_1-V_2\right\|_\infty,
\end{aligned}$$

where $\mathbf{1}_d = (1, ...., 1) \in \mathbb{R}^D$ and we have used that $\sum_{i=1}^N w_{N,i}(z,d) = 1$ by construction. The proof of $\Gamma$ being a contraction is analogous.

Next, we prove that $V_N(z,\lambda)$ is $s \geq 1$ times continuously differentiable under Assumption 2: We know that $\Gamma_N$ is a contraction mapping on $\mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$. But the set of $s \geq 0$ continuously differentiable functions $\mathbb{C}^s\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ is a closed subset of $\mathbb{B}\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$ and so the result will follow if $\Gamma_{N,\lambda}\left(\mathbb{C}_s\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D\right) \subseteq \mathbb{C}^s\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$. But for any $V \in \mathbb{C}^s\left(\mathcal{Z} \times \left(0, \bar{\lambda}\right)\right)^D$, it follows straightforwardly by the chain rule in conjuction with the stated assumptions that $\Gamma_N(V)(z,\lambda) = \sum_{i=1}^N G_\lambda\left(u_\psi\left(U_i;z\right)+\beta V_\psi\left(U_i;z,\lambda\right)\right) w_{N,i}\left(z\right)$ is $s \geq 0$ continuously differentiable w.r.t. $(z,\lambda)$. The proof of the Lipschitz property under Assumption 2(i) is similar and so left out. □

*Proof of Theorem 2.* We only show the result for $V_0$; the proof for the $V_N$ is analogous. Applying (3.8), the following holds for any $V$,

$$\begin{aligned}
\left|\Gamma(V)(z,0,d) - \Gamma(V)(z,\lambda,d)\right| \leq& \int \left|\max_{d\in\mathcal{D}}\left\{u(s',d)+\beta V(z',d')\right\} - G_\lambda\left(u(s')+\beta V(z')\right)\right| dF_S(s'|z,d) \\
\leq& \sup_{r\in\mathbb{R}^D}\left|G_\lambda\left(r\right) - \max_{d\in\mathcal{D}} r\left(d\right)\right| \int_{\mathcal{Z}\times\mathcal{E}} dF_s(ds'|z,d) \\
\leq& \lambda \log D.
\end{aligned}$$

The result now follows from the first part of Theorem A.1 with $\Psi_N(\cdot) = \Gamma(\cdot)(\cdot, \lambda_N)$. □

Our asymptotic analysis of $V_N$ proceeds in two steps: First, we develop a master theorem that delivers the desired result under a set of high-level conditions on the model and chosen importance sampler. The conditions are formulated to cover a wide range of different specifications, including both the case of $Z_t$ being continuously distributed or having countable support. Also, the master theorem allows for a wide range of the per-period utility functions and importance samplers. To state the high-level conditions, we recall the following definitions (see van der Vaart and Wellner, 1996): A class $\mathcal{F}$ of measurable functions mapping $U$ into $\mathbb{R}$ is called $P_U$-Glivenko-Cantelli if $\sup_{f\in\mathcal{F}} \left|\frac{1}{N}\sum_{i=1}^N f(U_i) - E[f(U)]\right| \to^P 0$ and it is called $P_U$-Donsker if $\sup_{f\in\mathcal{F}} \frac{1}{\sqrt{N}}\sum_{i=1}^N \{f(U_i) - E[f(U)]\} \rightsquigarrow \mathbb{G}$ in the space of all bounded functions from $\mathcal{F}$ to $\mathbb{R}$, where $\mathbb{G}$ is a tight Gaussian process.

**Theorem B.1.** *(i) Suppose that Assumption 1 is satisfied and the function classes* $\mathcal{W} := \{U \mapsto w_\psi(U;z) | \, z \in \mathcal{Z}\}$ *and*

$$\mathcal{G} = \left\{U \mapsto G_\lambda\left(u_\psi(U;z) + \beta V_0\left(\psi_Z(U;z),\lambda\right)\right) w_\psi(U;z) \middle| \, (z,\lambda) \in \mathcal{Z}\times\left(0,\bar{\lambda}\right)\right\} \tag{B.2}$$

*are $P_U$-Donsker. Then the first part of Theorem 3 holds.*

*(ii) Suppose furthermore that $V_N - V_0 \in \partial\mathcal{V}$ where $\partial\mathcal{V}$ is $P_U$-Glivenko-Cantelli with an integrable envelope function and*

$$\mathcal{G}' = \left\{U \mapsto \sum_{d\in\mathcal{D}} \dot{G}_{d,\lambda}\left(u_\psi(U;z) + \beta V_{\psi,0}(U;z,\lambda)\right) w_\psi(U;z) \middle| \, (z,\lambda) \in \mathcal{Z}\times\left(0,\bar{\lambda}\right)\right\}$$

*is $P_U$-Glivenko-Cantelli. Then the conclusions of Theorem 4 also hold.*

*Proof.* To show (i), we apply the first part of Theorem A.1 with $\Psi_N = \Gamma_N$, which is a contraction w.p.a.1, c.f. Theorem 1. First write

$$\Gamma_N(V)(z,\lambda) = \frac{\tilde{\Gamma}_N(V)(z,\lambda)}{W_N(z)}. \tag{B.3}$$

where

$$\tilde{\Gamma}_N(V)(z,\lambda) = \frac{1}{N}\sum_{i=1}^N G_\lambda\left(u_\psi(U_i;z) + \beta V_\psi(U_i;z)\right) w_\psi(U_i;z), \tag{B.4}$$

$$W_N(z) = \frac{1}{N}\sum_{j=1}^N w_\psi(U_i;z), \tag{B.5}$$

The Donsker condition on $\mathcal{G}$ and $\mathcal{W}$ now implies that

$$\sqrt{N}\left(\tilde{\Gamma}_N(V_0) - \Gamma(V_0), W_N - 1\right) \rightsquigarrow (\mathbb{G}_1, \mathbb{G}_2) \tag{B.6}$$

on $\mathcal{B}\left(\mathcal{Z}\times\left(0,\bar{\lambda}\right)\right)$, where $(\mathbb{G}_1,\mathbb{G}_2)$ is a Gaussian process, and so

$$\begin{aligned}\sqrt{N}\left\{\Gamma_N(V_0)-\Gamma(V_0)\right\} &= \sqrt{N}\left\{\tilde{\Gamma}_N(V_0)-\Gamma(V_0)\right\}-\Gamma(V_0)\sqrt{N}\left\{W_N-1\right\}+o_P(1)\\ &\rightsquigarrow \mathbb{G}:=\mathbb{G}_1-\Gamma(V_0)\mathbb{G}_2. \end{aligned}\tag{B.7}$$

In particular, $\left\|\Gamma_N(V_0)-\Gamma(V_0)\right\|_\infty=O_P\left(1/\sqrt{N}\right)$. We conclude from Theorem A.1 that $\|V_N-V_0\|_\infty=O_p(1/\sqrt{N})$.

To show the second part, we apply Theorem A.2. Weak convergence was derived above and it is easily seen that the influence function of $\Gamma_N(V_0)$ takes the form given in eq. (5.7) and so the Gaussian process $\mathbb{G}(z,\lambda)$ in eq. (B.7) has covariance kernel given in (5.6). The Frechet differential $dV\mapsto\nabla\Gamma_N(V_N)[dV]$ was derived in (4.14). It is a linear operator with $\|\nabla\Gamma_N(V_N)[dV]\|\leq\beta\|dV\|$ and so $dV\mapsto\{I-\nabla\Gamma_N(V_N)\}[dV]$ has a well-defined continuous inverse. Thus, what remains is to verify (iv) of Theorem A.2. This is done by showing uniform convergence of $dV\mapsto\nabla\tilde{\Gamma}_N(V_0)[dV]$ over $\mathcal{B}\left(\mathcal{Z}\times\left(0,\bar{\lambda}\right)\times\partial\mathcal{V}\right)$. But

$$\nabla\tilde{\Gamma}_N(V_0)[dV](z) = \frac{\beta}{N}\sum_{i=1}^{N}\sum_{d\in\mathcal{D}}\dot{G}_{d,\lambda}\left(u_\psi(U_i;z)+\beta V_{\psi,0}(U_i;z,\lambda)\right)dV_\psi(U_i;z,\lambda,d)\,w_\psi(U_i)\tag{B.8}$$

where $V_\psi(U;z,\lambda,d)\in\partial\mathcal{V}_\psi$ with

$$\partial\mathcal{V}_\psi=\left\{U\mapsto dV\left(\psi_Z(U;z),\lambda\right)\middle|\,(z,\lambda,dV)\in\mathcal{Z}\times\left(0,\bar{\lambda}\right)\times\partial\mathcal{V}\right\}$$

which is Glivenko-Cantelli since $\partial\mathcal{V}$ and $\{U\mapsto\psi_Z(U;z)|\,z\in\mathcal{Z}\}$ both have this property. Since $\mathcal{G}'$ is also Glivenko-Cantelli, it now follows from Theorem 3 in van der Vaart and Wellner (2000) that $\mathcal{G}'\cdot\partial\mathcal{V}_\psi$ is Glivenko-Cantelli as well which yields the desired result. □

*Proof of Theorem 3.* To show $\|V_N-V_0\|_\infty=O_P\left(1/\sqrt{N}\right)$, we verify the conditions of part (i) in Theorem B.1. First observe that $V_0(z,\lambda)$ is Lipschitz in $(z,\lambda)$, c.f. Theorem 1, and that $r\mapsto G_\lambda(r)$ is also Lipschitz uniformly in $\lambda\in(0,\bar{\lambda})$. Next, we show that $G_\lambda(r)$ is also Lispchitz w.r.t. $\lambda$ uniformly in $r$ by verifying that $\partial G_\lambda(r)/(\partial\lambda)$ is bounded uniformly in $\lambda\in(0,\bar{\lambda})$: Write

$$G_\lambda(r)=\lambda\log\left[\sum_{d\in\mathcal{D}}\exp\left(\frac{r(d)}{\lambda}\right)\right]=\max_{d\in D}r(d)+\lambda\log\left[\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)\right],$$

where $\bar{r}(d)=r(d)-\max_{d\in\mathcal{D}}r(d)\leq 0$, $d\in\mathcal{D}$, to obtain

$$\dot{G}_\lambda^{(\lambda)}(r)=\frac{\partial G_\lambda(r)}{\partial\lambda} = \log\left[\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)\right]-\frac{\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)\frac{\bar{r}(d)}{\lambda}}{\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)}.\tag{B.9}$$

Since $1\leq\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)\leq D$ and $-De^{-1}\leq\sum_{d\in\mathcal{D}}\exp\left(\frac{\bar{r}(d)}{\lambda}\right)\frac{\bar{r}(d)}{\lambda}\leq 0$ for all $\lambda>0$ and all

$r \in \mathbb{R}^D$, we conclude that $\left|\dot{G}_\lambda^{(\lambda)}(r)\right| \leq \log(D) + De^{-1}$ and so . Next,

$$\begin{aligned}
& \left|G_\lambda\left(u_\psi(U;z) + \beta V_0\left(\psi_Z\left(U,z\right),\lambda\right)\right) - G_{\lambda'}\left(u_\psi(U;z') + \beta V_0\left(\psi_Z\left(U,z'\right),\lambda'\right)\right)\right| \\
\leq \quad & \left|G_\lambda\left(u_\psi(U;z) + \beta V_0\left(\psi_Z\left(U,z\right),\lambda\right)\right) - G_{\lambda'}\left(u_\psi(U;z) + \beta V_0\left(\psi_Z\left(U,z\right),\lambda\right)\right)\right| \\
+ \quad & \left|G_{\lambda'}\left(u_\psi(U;z) + \beta V_0\left(\psi_Z\left(U,z\right),\lambda\right)\right) - G_{\lambda'}\left(u_\psi(U;z') + \beta V_0\left(\psi_Z\left(U,z'\right),\lambda'\right)\right)\right| \\
\leq \quad & C\left\{|\lambda - \lambda'| + \|u_\psi(U;z) - u_\psi(U;z')\| + \|V_0\left(\psi_Z\left(U,z\right),\lambda\right) - V_0\left(\psi_Z\left(U,z'\right),\lambda'\right)\|\right\},
\end{aligned}$$

and it now follows that under Assumption 2 together with the Lipschitz property of $V_0$ that $\mathcal{G}$ as defined in eq. (B.2) is Type IV class under $P_U$ with index 2 according to the definition on p. 2278 in Andrews (1994) which yields the first part of the theorem.

Next, we analyze $\partial V_N/\left(\partial z_j\right)$, $j = 1, ..., d_Z$. Since $\sup_{z\in\mathcal{Z}}\left|\sum_{i=1}^N w\left(S_i\left(z,d\right)|z,d\right)/N - 1\right| = O_P\left(1/\sqrt{N}\right)$, we replace $w_{N,i}\left(z,d\right)$ by $w_\psi\left(U_i;z,d\right)/N$ in the following. Now, taking derivatives w.r.t. $z_j$, $j = 1, ..., d_Z$, on both sides of eq. (4.1),

$$\frac{\partial V_N\left(z,\lambda\right)}{\partial z_j} = \nabla\Gamma_N(V_N)\left[\frac{\partial V_N}{\partial z_j}\right](z,\lambda) + \Gamma_{N,j}^{(z)}(V_N)\left(z,\lambda\right), \tag{B.10}$$

where $\nabla\Gamma_N$ was defined in (4.14) and

$$\begin{aligned}
\dot{\Gamma}_{N,j}^{(z)}(V_N)\left(z,\lambda,d\right) = \tfrac{1}{N} & \sum_{i=1}^N \sum_{d\in\mathcal{D}} \dot{G}_{\lambda,d}^{(r)}\left(u_\psi\left(U_i;z,d\right) + \beta V_{\psi,N}\left(U_i;z,\lambda\right)\right)\frac{\partial u_\psi\left(U_i;z,d\right)}{\partial z_j} w_\psi\left(U_i;z\right) \\
& + \frac{1}{N}\sum_{i=1}^N G_\lambda\left(u_\psi\left(U_i,d\right) + \beta V_{\psi,N}\left(U_i;z,\lambda\right)\right)\frac{\partial w_\psi\left(U_i;z\right)}{\partial z_j}.
\end{aligned}$$

where $\dot{G}_{\lambda,d}^{(r)}(r)$ was defined in (4.12). Similarly,

$$\frac{\partial V_N\left(z,\lambda\right)}{\partial \lambda} = \nabla\Gamma_N(V_N)\left[\frac{\partial V_N}{\partial \lambda}\right](z,\lambda) + \dot{\Gamma}_{N,j}^{(\lambda)}(V_N)\left(z,\lambda\right), \tag{B.11}$$

where

$$\dot{\Gamma}_{N,j}^{(\lambda)}(V_N)\left(z,\lambda,d\right) = \tfrac{1}{N}\sum_{i=1}^N \dot{G}_\lambda^{(\lambda)}\left(u_\psi\left(U_i;z,d\right) + \beta V_{\psi,N}\left(U_i;z,\lambda\right)\right) w_\psi\left(U_i;z\right),$$

and

$$\dot{G}_\lambda^{(\lambda)}\left(r\right) = \log\left[\sum_{d\in\mathcal{D}} \exp\left(\frac{r\left(d\right)}{\lambda}\right)\right] - \frac{\sum_{d\in\mathcal{D}} \exp\left(\frac{r(d)}{\lambda}\right)\frac{r(d)}{\lambda}}{\sum_{d\in\mathcal{D}} \exp\left(\frac{r(d)}{\lambda}\right)} \tag{B.12}$$

The mapping $dV \mapsto \nabla\Gamma_N(V_N)\left[dV\right]$ is a bounded linear operator with $\|\nabla\Gamma_N(V_N)\left[dV\right]\| \leq \beta\|dV\|$ and so

$$\frac{\partial V_N\left(z,\lambda\right)}{\partial z_j} = \left\{I - \nabla\Gamma_N(V_N)\right\}^{-1}\left[\dot{\Gamma}_{N,j}^{(z)}(V_N)\right](z,\lambda).$$

Thus,

$$\left\|\frac{\partial V_N}{\partial z_j}\right\|_\infty = \left\|\{I - \nabla\Gamma_N(V_N)\}^{-1}\left[\Gamma^{(z)}_{N,j}(V_N)\right]\right\|_\infty \leq \frac{\left\|\Gamma^{(z)}_{N,j}(V_N)\right\|_\infty}{1-\beta},$$

where,

$$\begin{aligned}\left\|\Gamma^{(z)}_{N,j}(V_N)\right\|_\infty \leq \tfrac{1}{N} & \sum_{i=1}^N \left\|\frac{\partial u_\psi(U_i;\cdot)}{\partial z_j}\right\|_\infty \|w_\psi(U_i;\cdot)\|_\infty \\ & + \frac{1}{N}\sum_{i=1}^N \left\{\|u_\psi(U_i;\cdot)\|_\infty + \beta\|V_N\|_\infty\right\}\left\|\frac{\partial w_\psi(U_i;\cdot)}{\partial z_j}\right\|_\infty.\end{aligned}$$

We know $\|V_N\|_\infty \to^P \|V\|_\infty$ and, under Assumption 2, we can appeal to the ULLN to obtain

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^N \left\|\frac{\partial u_\psi(U_i;\cdot)}{\partial z_j}\right\|_\infty \|w_\psi(U_i;\cdot)\|_\infty &\to^P E\left[\left\|\frac{\partial u_\psi(U;\cdot)}{\partial z_j}\right\|_\infty \|w_\psi(U;\cdot)\|_\infty\right], \\ \frac{1}{N}\sum_{i=1}^N \|u_\psi(U_i;\cdot)\|_\infty \left\|\frac{\partial w_\psi(U_i;\cdot)}{\partial z_j}\right\|_\infty &\to^P E\left[\|u_\psi(U;\cdot)\|_\infty \left\|\frac{\partial w_\psi(U;\cdot)}{\partial z_j}\right\|_\infty\right].\end{aligned}$$

We conclude that $\left\|\Gamma^{(z)}_{N,j}(V_N)\right\|_\infty$ and therefore also $\|\partial V_N/(\partial z_j)\|_\infty$ are bounded w.p.a.1. Similarly, it follows that $\|\partial V_N/(\partial\lambda)\|_\infty$ is bounded w.p.a.1. and so $V_N \in \mathbb{C}^1_r\left(\mathcal{Z}\times\left(0,\bar\lambda\right)\right)$ w.p.a.1 for some fixed $r<\infty$.

Finally, observe that $dV \mapsto \Psi_{N,j}(dV)(z,\lambda) = \nabla\Gamma_N(V_N)[dV](z,\lambda) + \Gamma^{(z)}_{N,j}(V_N)(z,\lambda)$ is a contraction mapping on $\mathbb{B}\left(\mathcal{Z}\times\left[0,\bar\lambda\right]\right)^D$ and so we can apply Theorem A.1. First, we expand each of the two terms w.r.t. $V_N$,

$$\begin{aligned}& \nabla\Gamma_N(V_N)[dV](z;\lambda) - \nabla\Gamma_N(V_0)[dV](z;\lambda) \\ = \;& \beta\sum_{i=1}^N \sum_{d_1,d_2\in\mathcal{D}} \ddot G^{(r)}_{\lambda,d_1,d_2}\left(u_\psi(U_i;z) + \beta\bar V_{\psi,N}(U_i;z,\lambda)\right)\left\{V_{\psi,N}(U_i;z,\lambda,d_2) - V_{\psi,0}(U_i;z,\lambda,d)\right\} \\ \times \;& dV(U_i;z,\lambda,d_1)\, w_\psi(U_i;z),\end{aligned}$$

where $\ddot G^{(r)}_{\lambda,d_1,d_2}(r) = \frac{\partial^2 G_\lambda(r)}{\partial r(d_1)\partial r(d_1)}$. It is easily checked that $\left|\ddot G^{(r)}_{\lambda,d_1,d_2}(r)\right| \leq C/\lambda$ for some $C<\infty$ and so the right hand side in the above equation is bounded by $C/\lambda\|V_N - V_0\|_\infty\|dV\|_\infty = O_P\left(\sqrt{N}/\lambda\right)$ for any given $dV \in \mathbb{B}\left(\mathcal{Z}\times\left[0,\bar\lambda\right]\right)^D$. By similar arguments, we can show that $\left\|\dot\Gamma^{(z)}_{N,j}(V_N) - \dot\Gamma^{(z)}_{N,j}(V_0)\right\|_\infty = O_P\left(\sqrt{N}/\lambda\right)$ and $\left\|\dot\Gamma^{(\lambda)}_{N,j}(V_N) - \dot\Gamma^{(\lambda)}_{N,j}(V_0)\right\|_\infty = O_P\left(\sqrt{N}/\lambda\right)$. Theorem A.1 now yields the second part of the theorem. □

*Proof of Theorem 4.* We verify the conditions in part (ii) of Theorem B.1 with $\partial\mathcal{V} = \mathbb{C}^1_r(\mathcal{Z})^D$ and $r<\infty$ given in Theorem 3. First, by arguments similar to the ones in the analysis of $\mathcal{G}$ in the proof of Theorem 3, $\mathcal{G}'$ is Glivenko-Cantelli due to the Lipschitz property of $V_0(\psi_Z(U;z),\lambda)$ and the other components entering the function set under Assumption 2. Second, $\mathbb{C}^1_r(\mathcal{Z})^D$ has finite Bracketing number according to Theorem 2.7.1 in van der Vaart and Wellner (1996) and so is also Glivenko-Cantelli.

The rate result is an immediate consequence of Theorem A.1 together with Assumption

3. For the weak convergence result, we use the decomposition (5.8) where $\left\|\hat{V}_N - V_N\right\|_\infty = O_p(\rho_{\Pi,K}) = o_P\left(1/\sqrt{N}\right)$ while the second term converges weakly according to Theorem 4. □

# C Additional numerical details for sieve method

## Chebyshev basis functions

Chebyshev polynomials of the first kind have well-known good properties for approximating functions on bounded intervals. Recall that Chebyshev polynomials are defined on $[-1, 1]$. We then choose $-\infty < z^{\min} < z^{\max} < \infty$ and define the $k$th basis function as follows for any $z \in \mathbb{R}$:

$$B_{c,k}(z) = \begin{cases} \cos\left((k-1)\arccos(T(z))\right), & |T(z)| \leq 1 \\ (\text{sign}(T(z)))^k, & |T(z)| > 1 \end{cases},$$

where $T(z) = 2\frac{z-z^{\min}}{z^{\max}-z^{\min}} - 1$ maps $z$ into the interval $[-1, 1]$. In particular, the basis functions are "truncated" and are set to one outside the interval $[z^{\min}, z^{\max}]$. This is done to avoid any erratic extrapolation. We then choose the grid points $z_1, ..., z_M$ in (4.7) as the Chebyshev nodes in order to minimize the presence of Runge's phenomenon. Thus, $M = K$ in this case.

## B-Splines

We use cardinal B(asis)-splines to form our B-spline spaces, so they are represented by a knot vector with equidistant entries $(0, \frac{1}{M+1}, \frac{2}{M+1}, \ldots, \frac{M}{M+1}, 1)$, and the Cox-de Boor recursion

$$\begin{aligned} \bar{B}_{i,0}(z) =& \begin{cases} 1 & \text{if } t_i \leq z < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ \bar{B}_{i,k}(z) =& \frac{z - t_i}{t_{i+k} - t_i}\bar{B}_{i,k-1}(z) + \frac{t_{i+k+1} - z}{t_{i+k+1} - t_{i+1}}\bar{B}_{i+1,k-1}(z). \end{aligned}$$

For interpolation purposes we use the so-called Universal (Parameters) Method by Tjahjowidodo et al. (2017). This amounts to choosing the $M$ grid points as the unique maximizers of all B-splines of degree $k \geq 1$, or any point if $k = 0$ in which case we set it to the first $K$ elements of the knot vector. The above are defined on the unit interval $[0, 1]$ and so the final basis functions are chosen as

$$B_{c,k}(z) = \begin{cases} \bar{B}_k(T(z)) & 0 \leq T(z) \leq 1 \\ (\text{sign}(T(z))) & \textit{otherwise} \end{cases}.$$

where now $T(z) = \frac{z-z^{\min}}{z^{\max}-z^{\min}}$.

Table 2: Coefficients on tensor product Chebyshev basis functions in the 2D model of engine replacement for $K = J^2 = 25$, $N = 200$.

| $J_1 \backslash J_2$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | -38.4713 | -4.4754 | 1.6176 | -0.256420 | -0.064960 |
| 2 | -4.4754 | 1.9662e-14 | -6.2341e-15 | -1.1318e-15 | 2.5392e-15 |
| 3 | 1.6176 | 7.2256e-15 | 5.6179e-14 | -2.0548e-14 | -3.3049e-15 |
| 4 | -0.2564 | -1.0110e-14 | -8.8673e-15 | 7.5672e-15 | -2.9838e-15 |
| 5 | -0.0649 | 4.0869e-15 | -1.5251e-14 | 3.4571e-15 | 1.4218e-15 |

Table 3: Coefficients on tensor product 2nd order B-Spline basis functions in the 2D model of engine replacement for $K = J^2 = 25$, $N = 200$.

| $J_1 \backslash J_2$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | -21.9800 | -27.1194 | -30.0583 | -31.1025 | -31.506 |
| 2 | -27.1194 | -32.2589 | -35.1977 | -36.2419 | -36.6455 |
| 3 | -30.0583 | -35.1977 | -38.1366 | -39.1808 | -39.5844 |
| 4 | -31.1025 | -36.2419 | -39.1808 | -40.2250 | -40.6285 |
| 5 | -31.5060 | -36.6455 | -39.5844 | -40.6285 | -41.0321 |

Table 4: Coefficients on the ten basis functions, and their products, upon convergence with $K = 50^2$.

| $J_1 \backslash J_2$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -52.200 | -5.070 | 2.700 | -1.170 | 0.478 | -0.145 | -0.006 | 0.038 | -0.024 | 0.007 |
| 2 | -5.070 | -1.560 | 0.135 | 0.225 | -0.057 | -0.023 | 0.013 | -0.006 | 0.004 | 0.001 |
| 3 | 2.700 | 0.135 | -0.176 | 0.032 | 0.046 | -0.021 | -0.004 | 0.005 | -0.002 | 0.001 |
| 4 | -1.170 | 0.225 | 0.032 | -0.087 | 0.019 | 0.019 | -0.012 | 0.002 | 0.001 | -0.001 |
| 5 | 0.478 | -0.057 | 0.046 | 0.019 | -0.038 | 0.010 | 0.009 | -0.008 | 0.002 | 0.000 |
| 6 | -0.145 | -0.023 | -0.021 | 0.019 | 0.010 | -0.018 | 0.005 | 0.005 | -0.004 | 0.002 |
| 7 | -0.006 | 0.013 | -0.004 | -0.012 | 0.009 | 0.005 | -0.009 | 0.003 | 0.002 | -0.003 |
| 8 | 0.038 | -0.006 | 0.005 | 0.002 | -0.008 | 0.005 | 0.003 | -0.005 | 0.002 | 0.001 |
| 9 | -0.024 | 0.004 | -0.002 | 0.001 | 0.002 | -0.004 | 0.002 | 0.002 | -0.003 | 0.001 |
| 10 | 0.007 | 0.001 | 0.001 | -0.001 | 0.0002 | 0.002 | -0.003 | 0.001 | 0.001 | -0.002 |

# Chapter 2
# Student Choices, Incentives, and Labor Markets Outcomes: The Case of Delayed Graduation

# Student Choices, Incentives, and Labor Markets Outcomes: The Case of Delayed Graduation

Bjørn Bjørnsson Meyer*

Patrick Kofod Mogensen*

**Abstract**

In this paper, we set up a dynamic choice model describing how various pecuniary and non-pecuniary incentives influence university students' decisions on part-time work, dropout, and delayed graduation. We estimate the model using Danish register microdata combined with administrative data from the country's largest university. Counterfactual simulations using the estimated model show that: (i) About half of the average delay in time-to-graduation can be explained by students following economic incentives to prepare for the labor market with work experience. The other half is due to a range of factors, such as income through part-time work and grants and the cost of effort for heavy course load. (ii) Cutting financial aid with one year reduces average time-to-graduation by 0.3 year, but also increases dropout.

*Department of Economics, University of Copenhagen

# 1 Introduction

Over the last couple of decades, there has been a significant expansion of post-secondary education in developed countries.[1] However, it has also become clear that the modern university system faces significant challenges. It has been stated that the university sector is in a completion crisis and that many institutions suffer from the *four-year myth*, meaning that postponing graduation is now more likely than not. Internationally, it is a widespread issue that too few students end up graduating, and that very few students graduate on time. Only 60 pct. earn a degree within eight years if we look at U.S. four-year bachelor's programs, and the average delay for graduates is 0.9 years.[2] Despite substantial institutional differences, such as free tuition and universal financial aid, the Danish university sector shows similar patterns in completion and delays. Only 70 pct. of Danish university students graduate in a three-year bachelor's program within five years, and the average time-to-graduation for a bachelor's plus master's degree is around six years instead of the expected five years.[3] Given the expected economic returns to post-secondary education, the typical view is that dropout and delayed graduation implies large public and individual costs.[4] Costs in terms of foregone earnings from high productivity workers and direct costs at the universities. This issue is a frequent topic in the policy debate on how to improve university education. Yet, there is little evidence on why students do not graduate on time and how various policy changes will affect student behavior.

In this paper, we develop an empirical dynamic model of student choices regarding part-time work, dropout, and course progression to study the process of university enrollment, and as a consequence, time-to-graduation. We construct our dataset from administrative records from the largest university in Denmark, University of Copenhagen (UCPH), coupled with individual register tax data on work hours and income. In particular, our unique data enables us to observe and model the sequential joint decision of study and part-time work intensity by semester as an optimal control problem where the utility maximization includes both life as a student and the income process after leaving university. In our model, part-time work is both a source of income and human capital investment. In contrast to existing studies, we explicitly account for the tradeoff between finishing on-time or later with more work experience and better grades. In our empirical framework, this channel shows to be important for graduating students. Since our model directly targets study intensity, it

[1] See Goldin and Katz (2009) and Aina et al. (2018).

[2] Complete College America (2014) and Planty and Hussar (2018) First time enrolled students in non-profit or public four-year institutions.

[3] UFM (2018a), UFM (2018b), and Statistics Denmark (2019).

[4] See among others, the handbook chapter by Deere and Vesovic (2006)

enables us to do policy simulations and evaluate the impact on time-to-graduation.

Based on our unique data, we first provide new descriptive insights on several key factors that are relevant for university enrollments and labor market outcomes. In addition to postponed graduation and high dropout rates, we document the following stylized facts: i) Students, who exit without a degree, have fewer passed courses and lower grades than their peers. ii) The chance of passing courses and getting higher grades is better for students enrolled in fewer courses, while there is little correlation between work hours and academic performance. iii) A gradual transition from university to the labor market with fewer credits, better grades, and more part-time work in late semesters. iv) Graduates experience significant income premiums compared to dropouts and graduates with higher grades and more work experience have more success in the labor market.

We incorporate all of the above features in our dynamic discrete choice model of part-time work and study intensity. The main building blocks are: i) Forward-looking students considering both future (dis)utility at university from work, courses, and grants, and utility from income after enrollment. ii) Labor market expectations depend on degree attainment, grades, and work experience while at university. iii) The increasing disutility of more work and study intensity.

Previous models like Arcidiacono et al. (2016) and Stinebrickner and Stinebrickner (2012, 2014) have quite parsimonious choice sets without work or study intensity choices.[5] Our data makes it possible to develop a more detailed model considering the tradeoffs between grades, part-time work, course progression, and labor market outcomes. In particular, we model course progression and delayed graduation explicitly. This policy dimension is increasingly relevant as there has been a substantial decline in on-time degree completion over time, see among others Turner (2004) and Bound et al. (2010). Existing models struggle to explain why students, despite the substantial economic incentives, delay completion. The existing literature (Lazear (1977), Kodde and Ritzen (1984), Oosterbeek and Van Ophem (2000), Carneiro et al. (2003), and more) tends to explain education beyond the seemingly income-maximizing level by non-pecuniary consumption value. In our model, we include the investment value of part-time work, as suggested by Ruhm (1997), Light (1999), and Scott-Clayton (2012), and a quantity-quality tradeoff in study intensity. Taking these channels into account, we find that a substantial part of the delay is due to an investment in human capital through work experience and better grades.

We show that the model can reproduce a distribution of time-to-graduation similar to the empirical one in simulations with the estimated parameters. Furthermore, the simulated

[5]The recent paper Joensen and Mattana (2017) uses Swedish data and model the part-time work decision. Their focus is on loan taking, and the data does not include grades or choice of study intensity.

choices are in line with the observed patterns of work hours and credits throughout the enrollment. We show by counterfactual simulations in the empirical framework that, i) About half of the average delay in time-to-graduation can be explained by the economic incentives to accumulate human capital for the labor market through part-time work. The remaining delay is due to various factors: the possibilities of income through part-time work and grants and the increasing cost of effort for credits, combined with the risk of failing exams and getting low grades when the course load is high. ii) Cutting financial aid with one reduces average time-to-graduation by 0.3 year but also increases dropout significantly. The first result is important since it reflects that a considerable part of the average delay is explained by students maximizing expectations of post-graduation income. This fact nuances the view that optimal policy leads to zero delayed graduation. In this light, it might be optimal to let students be enrolled longer and instead improve on academics and work experience. The fact that students can receive financial aid longer than on-time-graduation could be the primary explanation of why students postpone graduation. But we find that financial aid cuts have a somewhat limited scope as an instrument to reduce time-to-graduation, while it still imposes a possibly unintended consequence of more dropout. This is also an important contribution, as existing more parsimonious models would not lead to the same results as we include more elaborate channels for human capital accumulation.

The rest of the paper is organized as follows. Section 2 describes the institutional setting and data. Section 3 lays out the model. Section 4 show estimation results and model fit. Section 5 discusses various policy simulations based on the estimated model. Finally, section 6 concludes with some final remarks.

# 2 Institutional Setting and Data

In this section, we will first provide an overview of the students enrolled at UCPH, the faculty organization, tuition and subsidy structure, part-time work prevalence, and study progression. Then, we will take a closer look at some descriptive statistics and stylized facts that we find in the data and that our model should be able to capture.

## 2.1 Institutional Setting

UCPH has around forty thousand students and is the largest, and most selective in terms of student intake, of the eight Danish universities. A year of full-time studies corresponds to

60 credits. A bachelor's degree consists of 180 credits and a master's degree of 120 credits.[6] This means that the normed time-to-graduation of a bachelor's plus master's degree is five years.

Nonetheless, the rules regarding progression have been very flexible in our period, 2011-2015. To maintain enrollment in a bachelor's program, students were required to pass at least 60 credits within the first two years. If a student failed to meet this requirement, they could apply for an exception. Apart from this rule, the rules would vary between departments. All in all, there was a significant margin for part-time studies or even breaks for extended periods.

Programs at UCPH are field-specific from the first semester, implying that change of field requires dropout and re-enrollment in most cases. UCPH is organized in six faculties: Health, Social Science, Science, Humanities, Law, and Theology.[7] Faculties (except Law and Theology) are divided into smaller departments, where some departments offer several bachelor's and master's programs. In total, UCPH offers 73 bachelor's programs and 113 master's programs. The vast majority, 90.5 pct., of students who graduate with a bachelor's degree from UCPH, will also continue in a master's program at the same institution. Panel a) in Figure 1 shows how students accumulate credits and move towards graduation of bachelor's and master's degrees. Students who will graduate with a master's degree typically attain their bachelor's degrees in 6th and 7th semester and master's degrees between 10th and 14th semester. In this paper, we combine bachelor's and master's enrollments into one.

Programs typically initiate in the fall semester. The fall semester covers September to January and the spring semester February to June. Retake of spring semester exams often happens in August, and summer school courses are offered in July and August. For this reason, we include July and August in the spring semester, making it two months longer than the fall semester. We place courses in semesters by date of examination (see appendix A.1). A passed exam can either be passed in pass/no-pass grading or the grade 2 or above.[8] In general, grades are based on final examinations with no participation or midterm components. Most courses are 5, 7.5, or 10 credits.

Part-time jobs are prevalent among students at UCPH, both as an additional income source and viewed as important in terms of career prospects. For example, Klintefelt (2018) shows that part-time work correlates with better job opportunities. Work hours are typically spread out over the whole year, and few students engage in short term professional internships

[6]Like most European universities, UCPH follows the Bologna convention for program lengths and use credits measured in ECTS points, see European Union (2015).

[7]Theology is small and will henceforth be included in Humanities

[8]The Danish grading scale is usually normally distributed around 7, and includes the following marks with their letter grade conversions: $12(A), 10(B), 7(C), 4(D), 2(E), 00(Fx), -03(F)$.

Table 1: Summary Statistics

| *Individual level: (n=24,140)* | Mean | SD | *Semester level: (n=125,265)* | Mean | SD |
|---|---|---|---|---|---|
| Female | 0.60 | 0.49 | Registered credits | 23.78 | 12.98 |
| Age at entry | 22.19 | 5.12 | Passed credits | 20.80 | 13.05 |
| High school GPA | 8.27 | 2.01 | Semester GPA | 7.87 | 2.70 |
| Health | 0.07 | 0.26 | Hours per semester | 199.48 | 232.35 |
| Humanities | 0.34 | 0.47 | Hourly wage | 117.11 | 107.21 |
| Law | 0.15 | 0.36 | | | |
| Science | 0.25 | 0.43 | | | |
| Social Science | 0.18 | 0.38 | | | |

Note: Semester GPA is the weighted average of grades a student receives in a semester. In about 25 pct of semesters, the student does not pass any course or receive only a "passed" mark. The semester GPA variable is missing in these cases.

during the summer break. We observe hours worked and income monthly.

Tuition is free, and all students receive monthly financial aid amounting to DKK 5900 (USD 983[9]). In our sample period, students could receive financial aid for one additional year of enrollment on top of the program's normed time-to-graduation. Due to universal grants and no tuition fees, student loans play a relatively small role in Denmark.[10]

Our dataset contains 151,000 observations from spring 2011 to spring 2015, covering 29,200 students. We describe how we construct the dataset in appendix A.1. Table 1 shows that 60 pct. of UCPH students are female, and the average age first semester is 22.2 years. On average, students register 24 credits per semester, but only pass close to 21. On average, they work 199 hours per semester at an average wage of DKK 117/hour. This means that the study grant equals roughly 300 hours of work per semester.

## 2.2 Stylized Facts

We will now proceed to present six key stylized facts about study dynamics at UCPH. These findings suggest that there is important interactions between study intensity, examinations and finally labor market outcomes though the channels of human capital measured by degree attainment, grades and work experience.

1) *Delayed graduation.* Few students at UCPH finish on time in 10 semesters. Average time-to-graduation across all faculties is 12.7 semesters. As seen in panel b) in Figure 1 the most common time-to-graduation is 12 semesters.

[9]Exchange rate: 6 DKK = 1 USD

[10]Only 23 pct. of bachelor's students take a low-interest government loan in a given year. The average loan is only DKK 2200 per month, see UFM (2019).

2) *Passing courses and grades.* Controlling for previous academic performance the chance of passing courses and getting higher grades is larger for students enrolled in fewer courses, while there is little correlation between work hours and academic performance. The vertical distance between the lines in panel d) shows that there is a substantial gap between the amount credits students register and end up passing. See section 4.1 for more details and regressions.

3) *Dropout.* 33 pct. drop out within the first four years in a bachelor's program. 12 pct. of bachelor's graduates choose to exit instead of enrolling in a master's program. Dropout rates are considerably lower at the master's level where only 6 pct. drop out within four years of enrollment. Panel c) shows dropout happens throughout the semesters, but at a falling rate. Dropout rates are lowest around the timing of bachelor's graduation. Another take away is that the dropout rate increases from 1st to 2nd semester. This is in line with the idea that students respond to results from the first examinations in the end of 1st semester. Panels g) and h) show that dropouts are doing worse than average both in terms of grades and passing courses. The latter is also most predominant in the earlier semesters.

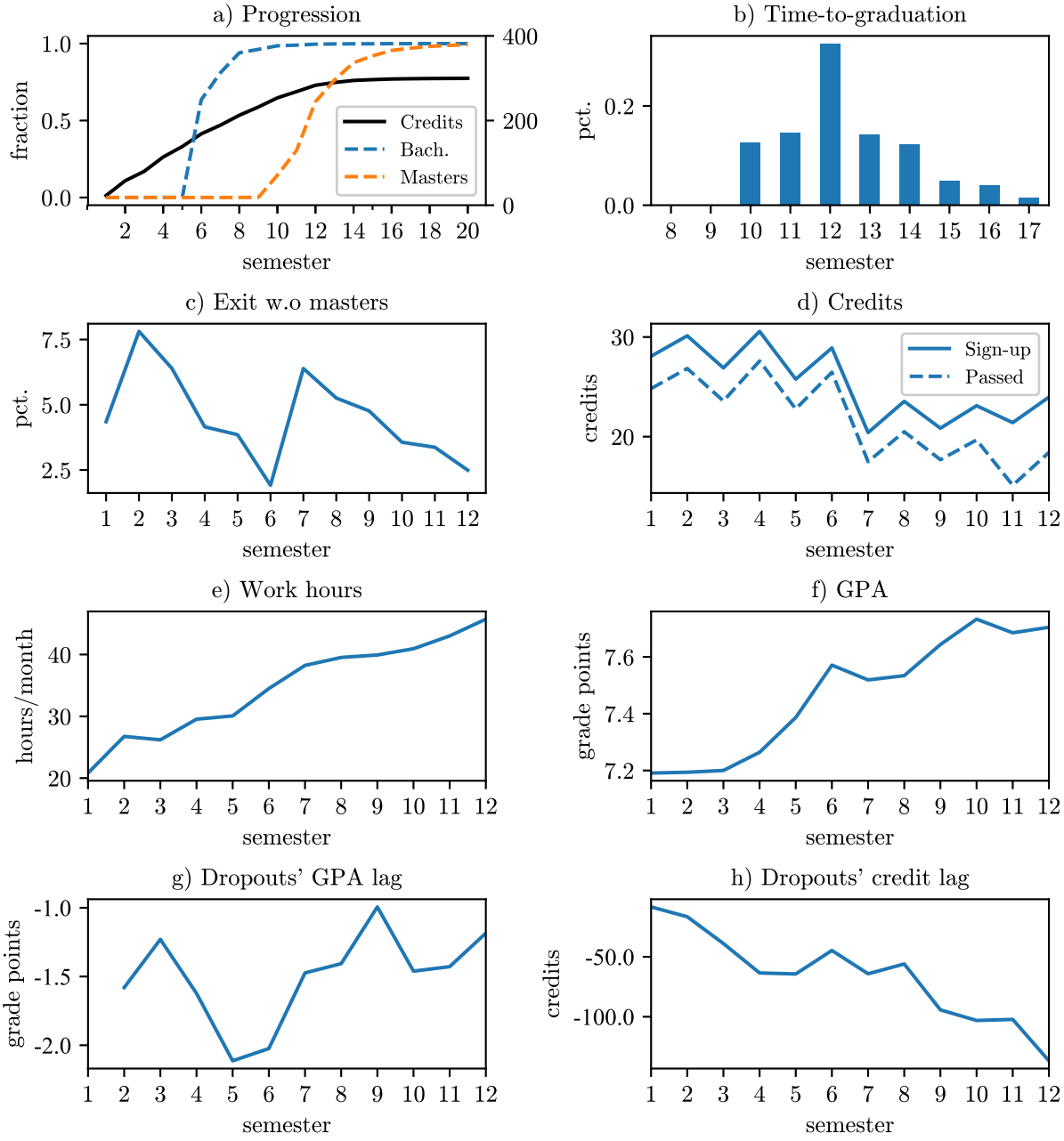
4) *A gradual transition from university to the labor market.* Credit progression slows down, but work intensity and GPA goes up over the enrollment duration. Panel d) shows decreasing credit progression over the enrollment and a seasonality pattern with more courses in the longer spring/summer periods. The relatively stable distance between the lines shows that the slowdown in progression is more due to fewer registered credits than more failed exams. The very last semesters is an exception to this as students tend to spend extra time on thesis writing. Panels e) and f) shows that the trend in work hours and GPA is opposite to credits. The sharpest increase (decrease) in work hours (credits) happens around the semesters where students progress to master's programs. There is some selection in the graphs as dropout happens throughout the enrollment and graduation is from semester 10 and onwards. But selection is not the main driver in these trends. Students who continue in master's programs experience an average slowdown of about six credits per semester compared to their bachelor's semesters. And students see an average increase in GPA of 0.2 grade points from their 6th to 10th semester.

5) *Graduates see large income premiums.* Figure 2 shows income profiles in general follows a reversed u-shape in tenure with income peaking between 25 and 30 years of experience. Students who drop out without a degree have around 40 pct. lower life time income despite their possibility to attain a degree at a later point in life. Earnings of students who leave UCPH with a bachelor's degree will eventually almost catch up to those of graduates with master's degrees. Earnings are substantially lower in the first 5 years or so reflecting the fact that many dropouts enters another educational program shortly after. In the following

years until earnings peak bachelor's seems to be about 5 years behind master's in terms of earnings. We see substantially heterogeneity across UCPH faculties for graduates but not so much for non degree dropouts. Relative premiums for degrees do also vary substantially. Faculty specific profiles are shown in Figure A.2.

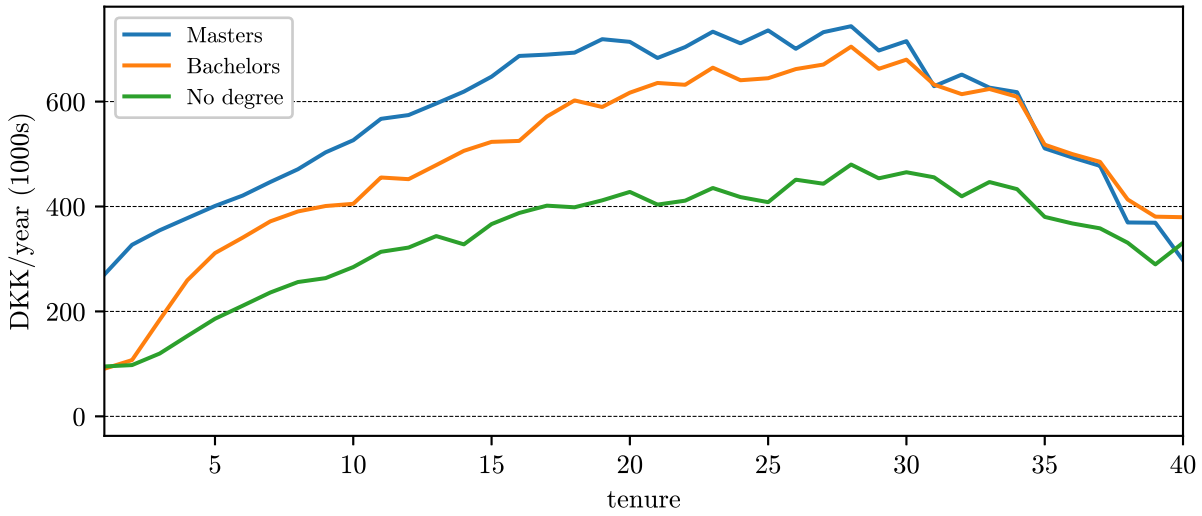
6) *Graduates with higher grades and more work experience earn more.* There is a sizable and significant positive correlation between grades and work experience from university and later income and employment chances. Figure 3 shows regression coefficients on both margins the first six years out of university for master's graduates. See section 4.2 for estimation details. The effect on employment is downward slopping over time while the effect on income is stable. In the first year after graduation one grade point higher GPA corresponds to 7.6 pct. higher income and 2.0 pp. higher probability of being employed. Similarly, 100 hours additional student work experience corresponds to 1.7 pct. higher income and and 0.8 pp. higher probability of being employed.

Figure 1: Enrollment Dynamics



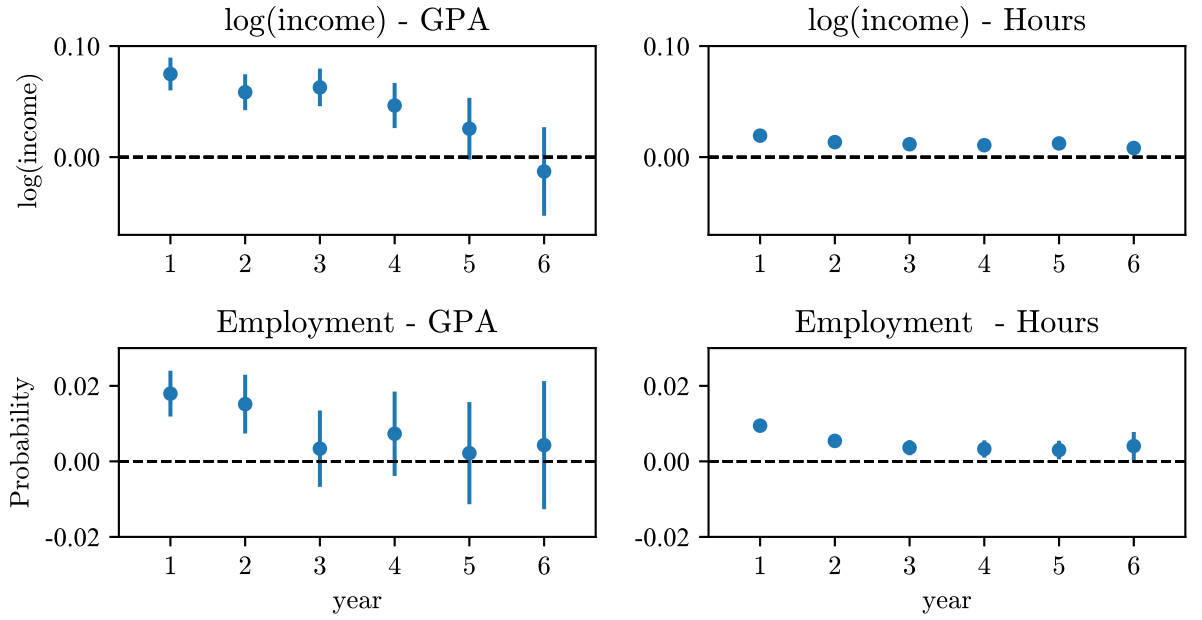

Note: a) Degree attainment over time for who complete a master's. Avg. accumulated credits on right axis. b) Bachelor's and master's degree combined to ten semesters normed graduation. c) Includes both early terminations of programs and bachelor's graduates who does not continue towards a master's. d) Enrolled and realized course credits. c)/d) The distribution of credits and hours is shown in appendix Figure A.1 e) Hours does not include self-employed. f) GPA is weighted by course credits. f)/g) The average distance between early terminating students and the faculty cohort averages in GPA and accumulated credits.

Figure 2: Income Profiles




Note: First time UCPH graduates and dropouts from earlier cohorts in Statistics Denmark's registers. Students might have attained degrees later. Wage and business income measured in 2015. Averages by each tenure year, calculated as 2015 minus year of graduation. There has only been bachelor's graduates (used to be combined programs) since 1994. Hence, avg. income after 21 years of tenure is extrapolated with the mean of the growth rates for master's graduates and no degree.

Figure 3: Grades and Work Experience Post-graduation




Note: Coefficient plots from income and employment regressions for master's graduates from 2011 and onwards. Controls for: age, gender, high school GPA and faculty. Graduates observed in up to first six years of income data (2012-2017). Std. errors are expected to increase as sample size becomes smaller (fewer cohorts). Right panel is overall marginal effects from a logistic regression. GPA on the Danish grading scale. Work experienced measured in 100s of hours.

# 3 A New Model of Study Progression

Our model of study progression includes both pecuniary and non-pecuniary incentives for student decisions to match the empirical findings from the section above. We emphasize the semester structure of choices and outcomes at university. The timeline of the model begins when students enroll at UCPH in a specific faculty at the university. This means that we do not model the decision to enroll or field choice. Students will then make choices regarding course enrollment and work hours every semester until they graduate or decide to drop out. The labor market is an absorbing state meaning that we do not consider re-enrollment in the model.[11] The life-time utility of students is split into two sequences of periods: a sequence of semesters enrolled at university followed by a sequence of years in the labor market. When a student enters the labor market, they take no further actions and receive labor market income according to their accumulated human capital and observable heterogeneity until retirement.

## 3.1 Dynamic Structure

Students enroll in university at age $A_0$ and stay there for a maximum number of semesters denoted $T_S$. At $T_S$ students will automatically enter the labor market with or without a degree. We set the number of semesters to 17 in the solutions used for estimation.[12] The student will receive labor market income from entry until a fixed retirement age, $A_{\max} = 60$. A student that leaves university after $t_s$ semesters has $A_{\max} - t_s/2 - A_0$ years left in the labor market. We include the time after the student exits university to be able to model the expected wage profiles of people with different characteristics when they leave university. The fixed retirement age means that students who spend additional years at university forgo years with labor market income. We abstract from modeling any saving or credit behavior.

Since the full model includes a sequence of student semesters followed by a sequence of labor market years, we need to define the sequential choice problems, or Bellman equations, for each of these cases. The model is solved recursively, but we present the equations in the order that the students face them. The yearly discount factor $\beta$ is set to 0.95 throughout the paper. Since our observations during enrollment come from semesters of varying length (seven and five months) we use differentiated values of $\beta$ to secure a consistent discount factor. This means that the discount factor is semester dependent and denoted $\beta_t$. For a spring semester, it is $0.95^{7/12}$ and for a fall semester, it is $0.95^{5/12}$.

[11] Potential earning gains from graduating later in life is captured in the income expectations for dropouts.
[12] In the data less than 2 pct. are enrolled longer than 17 semesters.

### 3.1.1 Study Progression

Students begin their education with initial conditions that are their age, gender, high school GPA, and faculty of enrollment. In each period they decide whether to continue their studies or drop out. We start by describing the student's problem given continued studies. Given continued enrollment, the student has to choose their study and work intensities. The credit choice is modeled as a target number of credits and is denoted $c_t^*$. At the end of the semester, the exams determine how many points the students accumulate. We denote the realized credit accumulation $c_t$ such that $c_t^* - c_t$ represents the number of failed credits. The stochastic process that models target versus realized credits is an ordered logit and will be described below. The work choice is given as the number of hours per month, $h$, they want to work. To make the model computational feasible we bin both credits and work choices into a reasonable sized discrete choice set of $(c^*, h)$ pairs. We use 6 equally spaced bins for credits ranging from 0 to 37.5 (full-time equals 30) and three levels of work hours, {none, low, high}.[13] The high-level structure of the enrolled student's problem is then given by

$$v_s(C, G, H, Z, \varepsilon, t) = \max_{c^*,\ h} \left[ u_s(t, c^*, h) + \varepsilon(\mathrm{s}, c^*, h) + \beta_t E\left(V_s(C', H', Z, \varepsilon', t+1)\right) \right] \qquad (1)$$

where $'$ denotes next period values of a variable. The variables in the equation are: credit stock ($C$), GPA level ($G$), stock of accumulated work hours ($H$), initial conditions ($Z$), and semester index ($t$). The taste shock vector ($\varepsilon$) has a component for each of the discrete choice combinations. We assume that these are IID extreme value type I random variables.[14] We assume conditional independence as is usual in the literature, see Rust (1987, 1988). Additionally, $u_s$ is the per period utility function for students. The initial conditions, Z, does not enter the instantaneos utility, but they affect the expectation operator through the laws of motion of states as we show below.

Once the student reaches 270 credits they have to write their master's thesis. In principle, this implies that students can choose between 0 (not writing their thesis this semester) or 30 (actively writing their thesis) credits. However, in the data we do not always observe that student sign their thesis contracts at the beginning of semesters. Therefore, it is difficult to match the data to a credit target for a student in our model. As a result, we deviate from the rest of the model by simply assuming they are perpetually enrolled to write their thesis until they drop out or graduate. They still accumulate work hours towards their human

[13] Low (high) is roughly 30 (60) work hours per month. See A.2 for more details.

[14] This implies the usual multinomial logit models, but it can be relaxed to nested models where study/work choices can be in a separate nest from dropout. See A.6.

capital stock, but the credit target is simply fixed at 30 credits.

If the student exits university, there are no more choices. The value of dropping out is captured by the following expression

$$v_d(H, Z, \varepsilon, t) = \alpha + \varepsilon(\mathrm{d}) + \beta_t V_d(H, Z, t+1) \tag{2}$$

where $\alpha$ is effectively a dropout intercept that captures the utility above and beyond the taste shock and expected utility of future income streams, $V_d$. This could include the option value of re-enrolling in a better student-field combination for example. We get back to the $V_d$ calculation below.

We can collect the two choice specific value functions, $v_s$ and $v_d$, in the value function in the following way, where the outer max represents the dropout decision

$$V_s(C, G, H, Z, \varepsilon, t \mid C < 300) = \max\Big\{v_s(C, G, H, Z, \varepsilon, t), v_d\Big\} \tag{3}$$

We condition on $C < 300$ because once the accumulated credits reach 300, the student succesfully exits university with their degree.

### 3.1.2 Labor Market

When the student enters the labor market there are no choices. The student will automatically enter the labor by graduation with a master's degree when $C_t = 300$ or if the limit for enrollment, $T_S$, is reached. Otherwise, the student can decide to drop out and enter the labor market at any point in time (with or without a bachelor's degree). We denote the age of the student in the time of exit as $A_e$. Since there are no choices in the labor market, the value function at the first period in the labor market, $V_e$, is:

$$V_e(C, G, H, Z, t) = \sum_{a=A_e}^{A_{\max}} E(\beta^a u_m(W_a(C, G, H, Z))) \tag{4}$$

The utility of money ($u_m$) is introduced to add decreasing marginal utility of money in the model. Specifically, we chose $u_m(x) = \sqrt{x}$.[15] To calculate the earnings profile $W$, we estimate a set of income regressions conditional on employment, and a set of logistic regressions on the employment probability for the first six years in the labor market. From the seventh year, we extrapolate with faculty-specific income growth rates. This process is described in more detail in the results section.

[15] This could be another function. We leave this for further research.

## 3.2 State Space

The state space consists of the following states: accumulated credits ($C$), current GPA ($G$), student work experience ($H$), and an index ($t$) for the number of semesters the student has been enrolled in. The first three play a role in the transition rules and the final wage the graduates can expect, and the index allows us to model seasonality of the semesters, and to limit the number of periods the students can get their universal study grants.

All states are discrete besides GPA. We model GPA as a continuous state, mostly for practical reasons. Grades and credit increments are naturally discrete, but the number of possible GPAs given a grading scale and the number of credits the student can accumulate per period would give rise to an extremely large state space, because the actual GPA is a weighted average of several levels of credits and semester grades.[16]

### 3.2.1 State Transitions

*Credits:* The accumulation of passed credits is straightforward:

$$C_{t+1} = C_t + c_t \tag{5}$$

However, to form expectations about this accumulation we need a model of $c_t$'s realizations given choices and states. Credit transitions are modeled as ordered logits with a separate regression for each level of registered credits. Hence students can reach no more than their target $c_t^*$, but there is a risk of passing fewer credits from failing exams. The regressions include work hours during the semester, start of period GPA, and a vector of student's characteristics as covariates. If students choose a target of zero, they trivially accumulate zero credits in that period. Degree attainment is a simple function of accumulated credits as students become bachelor's when they reach 180 credits and masters with 300 credits.

*Grades:* Students receive grades each semester with passed credits. These are combined with their current GPA to form next semester's GPA. The law of motion for GPA given the semester inventions is:

$$G_{t+1} = (G_t \cdot C_t + g_t \cdot c_t)/C_{t+1}, \tag{6}$$

where $g_t$ is the sum of grades divided by passed credits in the current semester. We see the increasingly persistent nature of GPA because $c_t$ becomes relatively smaller compared to $C_t$ as the students pass more and more courses. Note that $g_t$ and $c_t$ are stochastic. The process of $c_t$ is described above and we model $g_t$ using an ordered logit of possible per-semester GPAs

[16] We could also limit (discretize) the number of GPA states in each period as well, but we are concerned that this would make it hard to account for the effects of potential behavior in terms of lowering course registrations to do better per course.

(20 bins of half a grade point) including work hours and registered credits in the semester and a vector of students characteristics as covariates.

Since the GPA state only has discrete increments (by the combination of grades and realized credits), we could choose to keep this state discrete as well. However, the total number of combinations of realized credits and per semester grades quickly explodes, and for this reason, we model it as a continuous state.[17] We handle this continuous state by piecewise linear interpolation.

*Work experience:* Work experience is accumulated only after the 180'th credit has been accumulated. This is to emphasize the importance of the jobs students have during the master's degree. Empirically, we see that work experience accumulated during the master's enrollment has a larger effect on labor market outcomes. For computational reasons, we avoid counting bachelor's experience as an additional state. This means that $H_t = 0$ whenever $C_t < 180$. For $C_t \geq 180$ we have:

$$H_{t+1} = H_t + h_t \tag{7}$$

Alternative specifications are possible, among others: two types of human capital, continuously deprecating human capital, and stochastic increments (or decrements).[18] While different accumulation modeling strategies might add more flexibility or be more appropriate, they also increase the computational burden of the estimation. We leave this for further research.

## 3.3 The Utility of Being a Student

A large motivation for being a student is the expectation of higher wages upon graduation. However, since we model the dynamic progression, we also have to specify the instantaneous utility a student derives during each semester. This is a function of the current states and the discrete choices made. The total instantaneous utility $U$ can be decomposed into a base component $u$ and an element of the taste shock vector $\varepsilon$ given by the discrete choice. Starting with the case where the student decides to study, we have:

$$u_{\mathrm{s}}(t, c_t^*, h_t) = u_m(\mathrm{grant}(t) + y(t, h_t)) + \ell(c_t^*, h_t), \tag{8}$$

where grant$(t)$ is the study grant, $y(h_t)$ is the income from working while studying, $\ell(c_t^*, h_t)$ is non-monetary utility from being a student, and $u_m$ is the square root as described above

[17] Alternatively we could model a discrete Markov chain of several GPAs, but this blur the effect of the choices on future GPAs, and it complicates modeling sticky grades due to an increasing stock of credits.

[18] By stochastic decrements to $H$ we mean discrete deprecation at some probability.

to be the utility of money. In detail, we have that:

$$\text{grant}(t) = G \times I_{t\leq 12} \times m(t), \tag{9}$$

where $G$ is the level of the subsidy per month, $I_{t\leq 12}$ is an index function that is one for the first twelve semesters and zero after that. In the data $G =$ DKK 5900. The number of periods with eligibility for the subsidy is 12 in the data, but will be varied in the counter factual simulations. The function $m(t)$ scales the total amount according to the semesters such that uneven $t$'s are fall semesters with 5 months, and even $t$'s are spring semesters with 7 months. For wage income the students earn a wage proportional to their work hours:

$$y(t, h_t) = m(t) \times h_t \times w, \tag{10}$$

where $h_t$ is the chosen amount of semester work hours, and $w$ is the hourly wage. We use the median student wage calculated per faculty in the sample as a fixed parameter.[19] Across faculties, this amounts to somewhere between DKK 110 and 130.[20]

Lastly, the non-monetary utility consists of an intercept $\ell$ for university enrollment and power penalties for course load and work hours (both per month):

$$\ell(c_t^*, h_t) = \ell - \ell_c(c_t^*/m(t))^{\phi_c} - \ell_w h_t^{\phi_w} \tag{11}$$

The power functions allow for increasing marginal disutility of work and study effort, such that the last 7.5 credits are much harder to cope with mentally than the first 7.5, and that full-time work is much harder than part-time. We divide by length of the semester to accommodate for more time in spring semesters to study and pass courses. Notice, that students receive a quite generous subsidy. Depending on the faculty, monthly earnings are between DKK 3300 and 3900 for the low choice of hours (30) at the median wage. Which makes the subsidy 60 pct. of the total monthly income. If the student chooses the high amount of hours (60) the subsidy is still 40% of the total income.

## 3.4 Incentives and Mechanisms

In this section, we will describe some of the most important incentives students face in our model and how these mechanisms can lead to different behavior. The model is designed to

[19] We do not consider institutional earning restrictions for grant receivers since appendix A.5 shows that these are non-binding for most students.

[20] We could input their actual wage, or set up a model for wage offers. This would add another continuous state to the model, and is outside of the scope for the current analysis.

have multiple sources for prolonged enrollment, to let the data inform us on the relative importance of different channels.

*Stochatic delay.* Students are free to choose how many credits they register for, but are not guaranteed to pass them all. This means that if students sign up for 30 credits they will be delayed with high probability due to the significant chance of failing courses. How long depends on the demographics of the student and the realized grades. It is of course possible to register for 37.5 credits per semester, but that potentially places a very large disutility cost on the student - depending on the estimated parameter in the power function.

*Consumption value and grant income.* Policymakers sometimes emphasize that students simply stay because being a student is fun and attractive in and of itself. The student simply enjoys being enrolled, living the life of a university student. Working and studying a lot might subtract from this enjoyment that we'll refer to as the consumption value of enrollment. This is related to and possibly reinforced by, the generous student grant available to all full-time students. It's hard to quantify and identify consumption value in a model such as ours since it will end up consisting of many unobserved components, and all of them are not positive. Negative inherent features could be the stress of having to perform, balancing studies and work (above and beyond the actual work and study hours), and so on. However, the grant income enters directly in the model, and we can investigate the channel in counterfactual simulations.

*Improving labor market outcomes by better grades and work experience.* Another reason that students might rationally delay their graduation is to directly improve their grades and accumulated work experience. To balance the total disutility of effort per semester, the students might lower their registered credits to make room for more work, but they might also attend fewer classes to increase the likelihood of higher grades per course. Though it should be noted that more work hours do lower the probability of passing courses, and this might delay graduation too much.

*Transition into the post-university labor market.* The end goal is of course to finish the education and transition into the labor market outside of student work. Since we have already documented that there is a significant premium to both bachelor's and master's degrees, it is expected that students would want to balance the positive effects of more work experience with fewer years on the labor market before retirement, and so on. This means that it is likely not optimal to accumulate work experience indefinitely because it also lowers the possibility of passing courses even if the student registers for the 30 credits they are expected to register for.

For dropout, there are a few channels that are relevant to consider.

*Bad outlooks.* The attractiveness of dropout compared to the different study choices

is closely related to outlooks. Grades, credit stocks, work intensity, and study intensity are connected through the transition probabilities from period to period, and ultimately these help determine the returns to graduating in the end. A student who has stayed for three semesters but has had low realized grades and has failed many classes may then find themselves in a situation where graduation is still feasible, but it might require working very little (and accumulating very little work experience) and still result in a bad GPA, and so forth. Then, it might be better to drop out already in the third semester due to the low expected value of staying.

*Exit with a bachelor's degree.* The income prospects for exiting with a bachelor's degree compared to no degree or a master's varies between faculties. In some cases, there will be a bigger incentive to stay even if a student fails more classes than expected because they don't have to go for the full 300 points. We might then expect to see fewer exits just before the 180 credits required for a bachelor's degree is accumulated, and a spike just after.

*Unobserved preferences and field matches.* Students might not have a clear picture of the academic contents of a law degree versus a social science degree versus a physics degree. They might have some idea, but it's probably not too accurate. This means that we might observe people dropping out because their chosen field just doesn't fit their preferences after all. We don't model this beyond including a taste shock. However, it is clear that the taste shock also picks up things like mental stress, car accidents, and all the other things that might make students drop out even if all the features of the model tells us they should continue.

*Learning.* Our model contains learning about students' academic ability through the realization of grades that accumulate to their GPA. Students with a better high school GPA or a better GPA from previous semesters expect to do better in future courses both regarding the chance of passing and the grades received. Dynamic realizations of grades can change the outlook students are facing and hence the decision to drop out. This mechanism can be one explanation for students staying enrolled in some semesters before dropping out.

These are all potential explanations and channels that we will allow for in the model and investigate empirically.

## 3.5 Model Solution and Parameter Estimation

We solve the model using backward induction from the last period of studies. All choices are discrete, the maximization part is trivial - though the discrete state space is large, so each semester does take some time to solve.

We estimate parameters by maximum likelihood. A model with intended course progres-

sion, work intensity and drop out at the semester level would in many cases require simulated methods of moments, but since we have detailed data on student behavior, we are able to more directly match observed, individual choices with model predictions in terms of choice probabilities through maximum likelihood. We assume the shocks are extreme value type I. This means that we use the framework of Rust (1987), see A.6 for details.

With choice probabilities at hand, we can form the likelihood of observing the data. This simply amounts to summing up the log of the choice probabilities at the observed data. A less simple task is finding the estimates. In nonlinear estimation, it can sometimes be tricky to find initial parameters to start the optimization routine from, because some calculations can underflow if parameters are too far from those that generate patterns that roughly match data. We were able to generate initial values for our local searches using Adaptive Particle Swarm Optimization introduced in Yu and Guo (2013). For the local searches, we use a trust region method as described in Nocedal, Jorge and Wright (2006) and implemented in Mogensen and Riseth (2018). We use automatic differentiation (AD) to obtain gradients and Hessians. Specifically, we use the implementation of a forward mode automatic differentiation scheme developed in Revels et al. (2016). Having the Hessian available is helpful in both estimation and inference. The alternative is to use something like BFGS or BHHH. BFGS is popular in optimization, but in general, the inverse Hessian approximation can be very far from the true inverse Hessian, and the BHHH approximation is only valid near the maximum likelihood estimates.

# 4 Estimation Results

We can estimate the model sequentially by assuming the absence of type-specific unobserved heterogeneity and serially uncorrelated taste shocks. We use a two-stage approach. See Arcidiacono (2004) and Arcidiacono et al. (2016) for an application in our context, or Aguirregabiria and Mira (2010) for a more general survey. In the first stage, we estimate the parameters from the credit, grade, and income equations, $(\eta, \sigma_I, \pi, \psi, \delta)$. The second stage is devoted to the estimation of the flow utility parameters and option value from dropout, $(\ell_c, \ell_e, \ell_w, \phi_e, \phi_w, \alpha)$ by maximum likelihood, taking as given the first-stage estimates.

## 4.1 Academic Environment

*Course progression:* For each non-zero choice of course credit commitment, $c^*$, there is a risk of passing fewer credits. We model the passing probabilities by separate ordered logistic regressions for each credit level conditioning on student characteristics, GPA, and current

work hours. This secures that we get probabilities of passing less than registered credits for all possible student choices.

The left column of Table 2 shows the coefficients from the regression where $c^* = 30$ and Table A.3 in the appendix shows all levels of registered credits. In general, we find that the risk of not passing is lower for lower levels of registered credits. We also see that students with a high GPA are more inclined to pass all the credits they register. The coefficient on work hours is negative for all levels and tends to be larger for more registered credits. Nonetheless, coefficients are so small that the impact on passing probabilities is almost zero.

Some examples of predicted passing probabilities are: A male student with a 7.0 GPA registering for 15 credits has a 79 pct. chance of passing all credits. When registering 30 credits, the probability drops to 59 pct. For a student with a 12.0 GPA, the probabilities are 95 and 86 pct. respectively.

*Grades:* The relevant transition probabilities for the law of motion for the GPA in section 3.2.1 is the realizations of grades each semester, $g$. We discretize $g$ into 20 bins of half a grade point from 2 to 12. We model the realization of grades with a similar ordered logit, as in the section.[21] The right column of Table 2 shows coefficients. The coefficient for work hours is slightly positive but too small for work hours to have any real effect on grade probabilities.

## 4.2 Labor Market

To obtain the expected income profile ($W_a$) used in (4), we estimate equations both for earnings and the probability of employment by years of tenure using students exiting UCPH from 2011 to 2016.

Years of experience in the labor market is denoted $a$, faculty of study is $k$, and *no degree*, *bachelor's* and *master's* are indicators of degree attainment. A bar denotes the final values of a variable in the last semester at university. The earnings margin is estimated in the following linear log-income equation (conditional on positive earnings in the given year):

$$\begin{aligned}\log(\mathrm{Y}_a) =& \eta_{0,a,k} + Z\eta_{1,a} + \textit{master's} \times (\eta_{2,a}\bar{G} + \eta_{3an}\bar{H}) + \textit{bachelor's} \times (\eta_{4,a,k} + \eta_{5,a}\bar{G} + \eta_{6,a}\bar{H}) \\ &+ \textit{no degree} \times \eta_{7,a,k} + u_a, \end{aligned} \tag{12}$$

where $u_a \sim N(0, \sigma_a^Y I)$. There is no work experience term for students without a bachelor's as they have zero accumulated master's experience per definition.

Equations for the employment margin are estimated using logistic regressions with the

[21] In the model, students also need a GPA or expectation of a GPA when they enter university. We use the expected first semester GPA from an ordered logit with high school GPA, gender, and faculty of study as covariates.

Table 2: Course Results

| | (1) Credits | | (2) Grades | |
|---|---|---|---|---|
| | Est. | SE | Est. | SE |
| Registered credits | | | -0.020 | 0.001 |
| $GPA_{t-1}$ | 0.333 | 0.007 | 0.488 | 0.004 |
| Work hours (10s) | -0.003 | 0.001 | 0.006 | 0.000 |
| | | | | |
| *Controls:* | | | | |
| Female | 0.377 | 0.022 | 0.033 | 0.013 |
| High school GPA | 0.134 | 0.007 | 0.158 | 0.004 |
| Spring | 0.527 | 0.023 | 0.007 | 0.013 |
| Humanities | -0.102 | 0.043 | 0.390 | 0.024 |
| Law | 0.143 | 0.045 | 0.557 | 0.025 |
| Science | -0.375 | 0.041 | 0.451 | 0.024 |
| Social Science | -0.036 | 0.048 | 0.305 | 0.026 |
| N | 63,726 | | 93,822 | |

Note: Ordered logistic regressions. The unit of observation is semesters. Work hours are within the same semester as the course registrations. (1) Results when 30 credits are registered. Registered credits are not included as a covariate as there is a separate regression for each level. (2) Average semester grade is discretized into bins with increments of 0.5 grade points.

same right-hand side variables as in (12). Table 3 shows the results for the first year out of university.

GPA and work hours are significant and positive on both the employment and income margin. One grade point higher GPA (standard deviation of 1.65 grade points) corresponds to 7.6 pct. higher income and 2.0 pp. higher chance of being employed. Similarly, 100 hours (standard deviation of 1164 hours) additional student work experience corresponds to 1.7 pct. higher income and 0.8 pp. higher probability of being employed.

The results show that the highest income and employment chances are for Health graduates and the lowest for Humanities (same pattern as in Figure A.2). The mainly positive coefficients of bachelor's and no degree interactions with faculties cannot be interpreted as better labor market expectations without including the varying effects of work hours and GPA.

Figure 3 showed the GPA and work hours coefficients plotted for the first six years after graduation. The coefficient of GPA is decreasing over time. The coefficient of work hours is dropping the first few years, but later the size is more persistent. Since the high-intensity choice of work is 60 hours per month, master's students can accumulate a large stock of hours in the model. Combined with the persistent effect of work hours on the income profile, this makes work experience much more important than grades in this part of the model.

The coefficients obtained are used to calculate the earnings expectation. Letting $P_a$ denote the probability of being employed in a given year. We assume conditional independence[22] between $P_n$ and $Y_n$ and use the field-specific growth rates shown in Figure A.2 to extrapolate after the first six years in the labor market:

$$E[W_a] = P_a E[Y_a] = \frac{1}{1 + exp(-x\eta_{P_a})} exp(x\eta_{Y_a} + \sigma_Y^2/2) \tag{13}$$

For $a > 6$ :

$$E[W_a] = P_{\bar{6}} E[Y_{\bar{6}}] \pi_{a,k}, \tag{14}$$

where $\pi_{a,k}$ is the field-specific time trends including the employment margin where observed income is zero.

[22]This approximation of the income process is also used in Arcidiacono (2004).

Table 3: Labor market - First-year-out

| | (1) log Income | | (2) Employed | |
|---|---|---|---|---|
| | Est. | SE | Est. | SE |
| *Baseline (master's degree)* | | | | |
| Health | 11.938 | 0.106 | 0.141 | 0.029 |
| Humanities | 11.166 | 0.112 | 0.037 | 0.029 |
| Law | 11.784 | 0.105 | 0.102 | 0.03 |
| Science | 11.420 | 0.110 | 0.069 | 0.03 |
| Social Science | 11.608 | 0.111 | 0.075 | 0.031 |
| *Masters degree interactions* | | | | |
| Accumulated work hours (100s) | 0.017 | 0.001 | 0.008 | 0.001 |
| GPA | 0.076 | 0.007 | 0.020 | 0.003 |
| *Bachelor's degree interactions* | | | | |
| Health | -0.610 | 0.219 | 0.082 | 0.045 |
| Humanities | 0.871 | 0.192 | 0.119 | 0.035 |
| Law | 0.271 | 0.227 | 0.016 | 0.044 |
| Science | 0.682 | 0.196 | 0.064 | 0.036 |
| Social Science | 0.455 | 0.199 | 0.079 | 0.038 |
| Accumulated work hours (100s) | 0.022 | 0.002 | 0.007 | 0.001 |
| GPA | -0.030 | 0.018 | -0.001 | 0.003 |
| *No degree interactions* | | | | |
| Health | -0.810 | 0.129 | 0.065 | 0.031 |
| Humanities | 0.774 | 0.125 | 0.107 | 0.026 |
| Law | 0.282 | 0.138 | 0.046 | 0.029 |
| Science | 0.548 | 0.126 | 0.073 | 0.026 |
| Social Science | 0.381 | 0.139 | 0.072 | 0.029 |
| *Controls* | | | | |
| Female | -0.112 | 0.020 | 0.002 | 0.005 |
| High school GPA | -0.006 | 0.007 | 0.001 | 0.001 |
| Age | 0.012 | 0.003 | -0.003 | 0.000 |
| *Residual MSE* | 1.180 | | | |
| N | 12,763 | | 14,366 | |
| R2 | 0.269 | | | |

Note: Graduates from 2011 and onwards. Employed if income $> 0$. Right column is a logistic regression. First-year-out is Jan 1st through Dec 31st the year after graduation. We also control for month of exit to avoid tenure effects for students exiting early in the year. Work hours are accumulated while enrolled at master's level. Work hours from before 2008 imputed from earnings data.

## 4.3 Estimating Utility Parameters

We find that there is quite some heterogeneity in the student behavior between faculties, see appendix A.3. Some of this is arguably due to faculty specific heterogeneity we already include in various parts of the model, such as the academic environment and labor market expectations. Nonetheless, we also find it plausible that utility coefficients also vary between faculties, and even between programs within the same faculty. This could be due to more difficult courses making it more costly to take credits or better opportunities for enjoyable part-time jobs in some programs.

To be able to fit this dimension of heterogeneity, we estimate faculty-specific utility coefficients. We have limited our model estimation to the faculties of Law and Social Science. All students at the Faculty of Law are in the same program, making it the most homogeneous faculty. Furthermore, there is a well-defined path from a bachelor's in law to a master's in law and finally, a close link to a specific profession. On the other hand, Faculty of Social Science includes a range of different programs such as economics, anthropology, and psychology. Here course content, teaching, and the following labor markets vary widely. On average social science students have a one point higher high school GPA. Law students tend to have more part-time work, shorter time-to-gradation, and better labor market outlooks than social science students and the UCPH average (see appendix A.3).

Table 4 shows the model estimates. The estimates of $\ell_c$ and $\ell_w$ are positive, which is in line with our expectation that study effort and work effort are indeed disutilities. They enter with negative signs in the utility functions, so higher effort in terms of credits and work hours means more disutility. The exponents are somewhere around the quadratic case with the work exponent slightly above for Social Science (at 2.34), and the study exponent slightly below for Law (at 1.64). The cost of work hours is higher for Social Science as both parameters exceed their Law counterparts. For credits, it is more ambiguous since $\ell_c$ is largest for Social Science, but the curvature parameter $\phi_c$ is largest for Law.

The semester utility intercepts, $\ell$, are estimated to be negative and close to each other. We have to keep in mind here that this is above and beyond the utility from grant and wage income. The dropout intercepts, $\alpha$, are positive and somewhat larger for Social Science. This indicates that there are advantages to dropping out that are not captured in our pecuniary dropout value given by the wage equation only.

## 4.4 Model Fit

Given the parameter estimates in Table 4, we now examine how well the model is able to replicate the trends in data. Figure 4 shows our model simulation overlayed with the data.

Table 4: Utility Coefficients

| | Social Science | Law |
|---|---|---|
| *Intercepts:* | | |
| $\ell$ | -0.43 | -0.39 |
| $\alpha$ | 0.69 | 0.40 |
| *Credit parameters:* | | |
| $\ell_c$ | 0.04 | 0.08 |
| $\phi_c$ | 1.96 | 1.64 |
| *Work hours parameters:* | | |
| $\ell_w$ | 4.45 | 3.39 |
| $\phi_w$ | 2.34 | 1.94 |

Note: Parameters are estimated separately but use the same first stage estimations.

In general, the model predicts the trends in data fairly well. We get a similar shape of time-to-graduation distribution with a mean very close to the data. In terms of registered and passed credits, we see more credits in longer spring semesters, a relative constant credit failing rate, and a slow down in intensity around the shift to master's programs (driven by incentives invest in work experience) all in conjecture with the data. Despite a too large acceleration during the transition into master's semesters, we are reasonably close to the average choice of work hours.

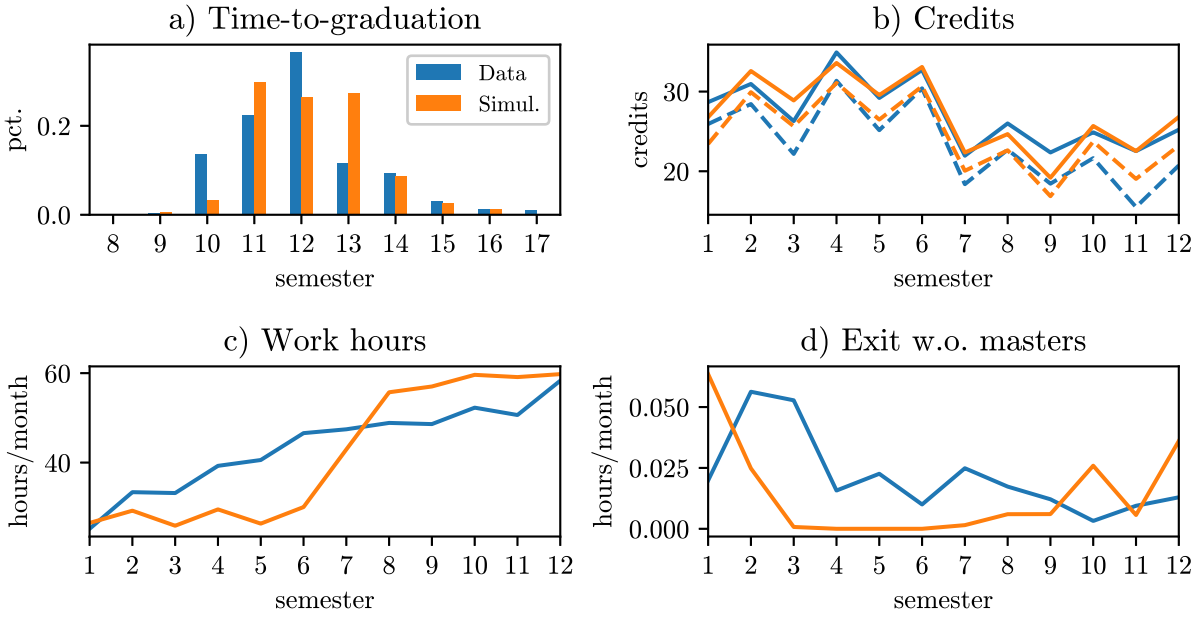
However, the model is somewhat limited in its ability to replicate the dynamic pattern of dropout. We simulate too high dropout rates early and late in the enrollment and almost zero dropouts in the middle. The high dropout rate in the first semester can be interpreted as students being too sure about future outcomes early in our model. An explanation for this is that dropout, in many cases, is driven by other reasons than pure labor market expectations. As already mentioned earlier, Stinebrickner and Stinebrickner (2014) find that students updating labor market expectations can only account for 22.5 pct of the observed dropout leaving lots of space for explanations like changing preferences and interests throughout the enrollment.

# 5 Counterfactual Simulations

Given that the model matches the data reasonably well, we can use it to do two counterfactual simulations.

*(I):* No human capital investment in work experience. In this simulation we ask the question, how much of delayed in time-to-graduation can be explained by students postponing

Figure 4: Model Fit - Faculty of Law




Note: Simulation with model parameters of students at the Faculty of Law. b) Solid lines are registered credits. Dashed are passed credits. See the model fit for the Social Science faculty in Figure A.3 in the appendix.

with the aim of improving labor market outcomes by work experience? We impose this on the model by setting all work experience at graduation to the mean amount of hours students graduate within the sample. This means that choices during enrollment do not affect work experience but students still earn income from part-time work. The results can be seen in column (I) in Table 5. The average time-to-graduation effect is a reduction of 1.4 (1.3) semesters for Social science (Law). This means that they delay (excess of ten semesters) is reduced by 53-59 pct. Hence, our results indicate that a significant part of the delay is due to rational human capital accumulation on the students' side. We do notice a significant decrease in drop out behavior as well. In that regard, we should keep in mind that by not having to earn the work experience, we are saving the students from a lot of work effort disutility. This means that the students who would choose to drop out in the baseline now have a much more attractive (and unrealistic) path from an unfortunate state realization to graduation.

*(II):* Reduction in financial aid. One persistent policy proposal in the debate is to reduce the time students are eligible to receive financial aid. In the baseline model and the data, students can receive grants for six years. In this counterfactual simulation, we reduce this period by one year, corresponding to the normed time of a bachelor's plus master's degree. The results can be seen in column (II). The effect on average time-to-graduation is a reduction by 0.7 (0.6) semesters for Social science (Law) or a relative reduction of the delay by 27 pct.

Table 5: Counterfactuals

| | Baseline | (I) | (II) |
|---|---|---|---|
| *Law:* | | | |
| Time-to-graduation | 12.2 | 10.9 | 11.6 |
| Exit w/o. masters degree (pct.) | 15.5 | 0.5 | 25.0 |
| | | | |
| *Social science:* | | | |
| Time-to-graduation | 12.6 | 11.2 | 11.9 |
| Exit w/o. masters degree (pct.) | 26.5 | 0.4 | 35.4 |

Note: Baseline is calculated on a dataset simulated with the fitted model parameters. (I) and (II) are calculated on counterfactual datasets.

for both faculties. However, we also see 8.9-9.5 pp. more students are exiting without a master's degree.

# 6 Concluding Remarks

Large extensions in time-to-graduation for university degrees persist in most developed countries, including Denmark. To unfold the incentives and choices resulting in delays, we use a novel dataset and estimate a dynamic model of students' choices regarding course load, part-time work, and dropout. We model the dimension of choosing course load directly and use more refined channels of human capital formation through work experience and grades, which has not been included in previous models in the literature. Our model is able to capture quite a lot of the observed variation in the actual study behavior with a relatively parsimonious utility specification without unobserved heterogeneity.

Our first main result is, that about half of the delay to graduation can be explained by students choosing to prolong enrollment and accumulate more human capital through part-time work experience. This finding brings nuance to the debate about delayed graduation. Even with the narrow goal of university policy to maximize university students' lifetime earnings, it can be the case that optimal time-to-graduation is longer than the normed five years. Hence, future policy should take these dynamics into account and be specific about the end goals like cutting direct costs of tuition and financial aid, or maximizing labor market outcomes for students.

Our second main result is, that cutting financial aid by one year reduces time-to-graduation by one-third of a year. The reduced pass-through of aid cuts highlights that other factors such as failing exams, part-time work, grades, and cost of effort are also important explanations of

delays. The overall tradeoff in terms of reducing financial aid is whether negative effects like more dropout and fewer applicants outweigh the reduction in costs and time-to-graduation. In the data and the model, dropout mostly occurs and increases among students with worse academic performance and hence worse academic and labor market outlook. Hence, it would be necessary to investigate this heterogeneity further in the model to make statements about the labor market costs of forgone earnings for these individuals. Before venturing into such an analysis, we believe the model would benefit from being extended with more student heterogeneity in the following two directions. First, adding Bayesian learning for students about their unobserved academic ability would make the learning process more realistic, and could imply that more students with bad academic realizations would update expectations downwards and drop out. Secondly, our model could be improved with a more general form of unobserved heterogeneity in student ability and preferences. In particular, adding heterogeneity, potentially tied to student characteristics, in students' costs of effort for course load seems relevant.

An additional danger, policy should account for, is that more drop out among worse-performing students reflects incoming students' academic preparation, and more inequality in educational attainment is a potential consequence.

# References

Aguirregabiria, V. and P. Mira (2010). Dynamic discrete choice structural models: A survey. *Journal of Econometrics 156*(1), 38–67.

Aina, C., E. Baici, G. Casalone, and F. Pastore (2018). The Economics of University Dropouts and Delayed Graduation: A Survey. Technical report.

Arcidiacono, P. (2004). Ability sorting and the returns to college major. *Journal of Econometrics 121*(1-2), 343–375.

Arcidiacono, P., E. M. Aucejo, A. Maurel, and T. Ransom (2016). College Attrition and the Dynamics of Information Revelation. *NBER Working Paper* (No. w22325).

Bound, J., M. F. Lovenheim, and S. Turner (2010). Why have college completion rates declined? an analysis of changing student preparation and collegiate resources. *American Economic Journal: Applied Economics 2*(3), 129–157.

Carneiro, P., K. T. Hansen, and J. J. Heckman (2003). Estimating distributions of treatment effects with an application to the returns to schooling and measurement of the effects of uncertainty on college choice. *International Economic Review 44*(2), 361–422.

Complete College America (2014). Four-Year Myth. Technical report, Complete College America.

Deere, D. R. and J. Vesovic (2006). Chapter 6 Educational Wage Premiums and the U.S. Income Distribution: A Survey. *Handbook of the Economics of Education 1*(06), 255–306.

European Union (2015). ECTS Users' Guide. Technical report, European Union.

Goldin, C. D. and L. F. Katz (2009). *The Race between Education and Technology*. Harvard University Press.

Joensen, J. S. and E. Mattana (2017). Student Aid, Academic Achievement, and Labor Market Behavior. *IFAU Working paper*, 1–45.

Klintefelt, T. (2018). Studiearbejde øger chancerne for beskæftigelse. Technical Report September, Dansk Industri.

Kodde, D. A. and J. M. M. Ritzen (1984, nov). Integrating Consumption and Investment Motives in a Neoclassical Model of Demand for Education. *Kyklos 37*(4), 598–608.

Lazear, E. (1977). Education : Consumption or Production? *85*(3), 569–598.

Light, A. (1999). High school employment, high school curriculum, and post-school wages. *Economics of Education Review 18*(3), 291–309.

Mogensen, P. K. and A. N. Riseth (2018). Optim: A mathematical optimization package for Julia. *Journal of Open Source Software 3*(24), 615.

Nocedal, Jorge and Wright, S. (2006). *Numerical optimization*. Springer Science \& Business Media.

Oosterbeek, H. and H. Van Ophem (2000). Schooling choices: Preferences, discount rates, and rates of return. *Empirical Economics 25*(1), 15–34.

Planty, M. and W. Hussar (2018). The Condition of education 2018. Technical report, U.S. Department of Education.

Revels, J., M. Lubin, and T. Papamarkou (2016). Forward-Mode Automatic Differentiation in Julia.

Ruhm, C. J. (1997). Is high school employment consumption or investment? *Journal of Labor Economics 15*(4), 735–776.

Rust, J. (1987). Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher. *Econometrica 55*(5), 999–1033.

Rust, J. (1988). Maximum Likelihood Estimation of Discrete Control Processes. *SIAM Journal on Control and Optimization 26*(5), 1006–1024.

Scott-Clayton, J. (2012). What explains trends in labor supply among U.S. Undergraduates? *National Tax Journal 65*(1), 181–210.

Statistics Denmark (2019). StatBank. Education and Knowledge. Technical report, Statistics Denmark.

Stinebrickner, R. and T. Stinebrickner (2014). Academic performance and college dropout: Using longitudinal expectations data to estimate a learning model. *Journal of Labor Economics 32*(3), 601–644.

Stinebrickner, T. and R. Stinebrickner (2012). Learning about Academic Ability and the College Dropout Decision. *Journal of Labor Economics 30*(4), 707–748.

Turner, S. E. (2004). *Going to College and Finishing College: Explaining Different Educational Outcomes*. Number September. NBER.

UFM (2018a). Frafald blandt akademiske bachelorstuderende. Technical report, Uddannelses- og Forskningsministeriet.

UFM (2018b). Udviklingen i studietiden. Technical report, Uddannelses- og Forskningsministeriet.

UFM (2019). Oplysninger om tildeling af SU for 2018. Technical report, Uddannelses- og Forskningsministeriet.

Yu, X. and J. Guo (2013). A novel adaptive particle swarm optimization. *Journal of Engineering Science and Technology Review 6*(2), 179–183.

# A Appendix

## A.1 UCPH Data

Data from UCPH is merged by social security number to population registers at Statistics Denmark. We use tax data based registers (*IND,E-indkomst*) on income and hours worked

per month, demographics (*BEF*), and educational registers (*UDDA, UDG*) for high school GPA and cross-checking UCPH enrollments.

We have data from UCPH from fall 2011 and onwards. In our main sample, we exclude all semester observations where students are affected by the Study Progress Reform, which mandated full-time study intensity. This means semesters from fall 2014 for entering students and fall 2015 for all students. Age, gender, and high school GPAs are merged from DST's population registers (BEF and UDG). There is almost complete coverage for the demographic variables, but high school grades are missing for all students without a regular Danish high school examination, namely international students. We furthermore restrict our sample in several dimensions. The sample restrictions are described in Table A.1.

The majority of courses at UCPH last half or whole semesters. Some programs allow for a two-semester master's thesis. We divide these over two semesters. We place courses in semesters by date of examination. A course with an examination within the first month of a semester is counted towards the previous semester, as it will most likely be a re-examination. Students were allowed to change course registrations early in the semesters, but we only consider final subscriptions. Once exam registrations are locked, not passing an exam results in a "missed try" of which students have a maximum of three per course. Numerical grades count towards the GPA, which is weighted by course credits.

We place exits in semesters by the registered date. If the exit date is within the first month of a semester, we replace the exit decision in the previous semester. If an exit date occurs in a semester where course credits also are passed, we replace it in the following semester.

## A.2 Binning Credits and Work Hours

To reduce the state space and the number of choices in each period, we bin credits and work hours. In the actual data, there's a wide variety of ways you can choose combinations of courses with varying credit size. Some write extended essays for half a point, some courses are more comprehensive and count for ten or even 20 credits. The combinations of these many, unequally spaced bins give rise to an enormous state space, especially in the later semesters where courses from many semesters are accumulated. Therefore we use an equal spacing of 7.5 credit points per semester: 0.0, 7.5, 15, 22.5, 30, and 37.5. These choices are representative in the sense that most common sizes of courses (and thesis projects) are dividable with 7.5. We allow for more than 30 points (full-time studies), as this is also possible with either an extra regular course or a summer school course in spring semesters. Alternative numbers are binned to the closest value up or down. The binned state space of

Table A.1: Sample Size

| | Sample restriction | unique IDs |
|---|---|---|
| i) | Pre-reform semesters (fall 2011-spring 2015) | 63298 |
| ii) | Not enrolled before 09/01/2000 | 58694 |
| iii) | No overlapping enrollments | 56052 |
| iv) | No shorter or longer programs | 46227 |
| v) | No spring semester enrollments | 45629 |
| vi) | No transfer or reentrying students | 40368 |
| vii) | No unexplained graduation stocks | 36108 |
| viii) | High school GPA in register | 31333 |

Note: ii) Excluding some students with very long passive enrollments. iii) Students with simultaneous enrollments in different programs are excluded. iv) Some programs have other credit requirements for graduation. Notably, shorter professional programs and the longer master's degree in medicine v) Having enrollments in both spring and fall makes the model more computationally demanding. vi) Excluding all students with transferred credits from previous enrollments at UCPH or other universities. Re-enrollment in the same program can disappear in the UCPH data. Hence we cross-check with DST's education register (UDDA). We include enrollments where a bachelor's graduate continues in a master's program within a year. vi) Excluding students who graduate with a different credit stock than expected. iv)-vi) could potentially be incorporated in the sample and model setup.

accumulated passed credits is all increments of 7.5 from 0 to 300. We bin the actual credit stock in each semester to the closest value up or down.

We limit students to not enroll in more credits than they have left plus one increment and make this change this in the dataset too.

We also bin the distribution of work hours. We bin to a three-point distribution consisting of zero hours and low/high-intensity part time employment. High (low) corresponds to 60 (30) 60 hours per month. Positive work hours per month are binned to the closest value up or down. The accumulated work experience is the sum of the binned work hours choices each semester corrected for the length of the semesters (this is procedure is chosen for computational reasons).

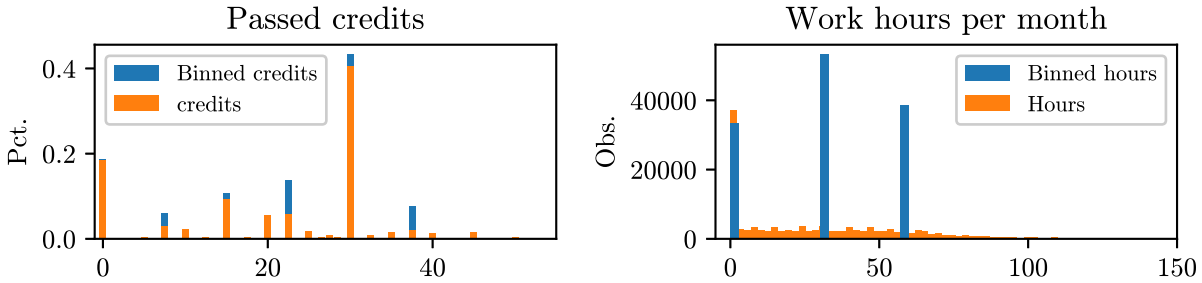
Figure A.1 shows the distributions of work and credits per semester in the dataset before and after implementation of the described binning procedure.

## A.3 Faculty Heterogeneity

Figure A.2 shows the income profiles for students exiting with a master's, bachelor's, and without a degree split by faculties all five faculties.

Table A.2 shows summary statistics for the faculties Social Science and Law separately.

Figure A.1: Bins of Choice Space




Note: Credits in the left panel are all observed levels. Hours per month in the right panel is a histogram with 140 bins.

Table A.2: Summary Statistics by Faculty

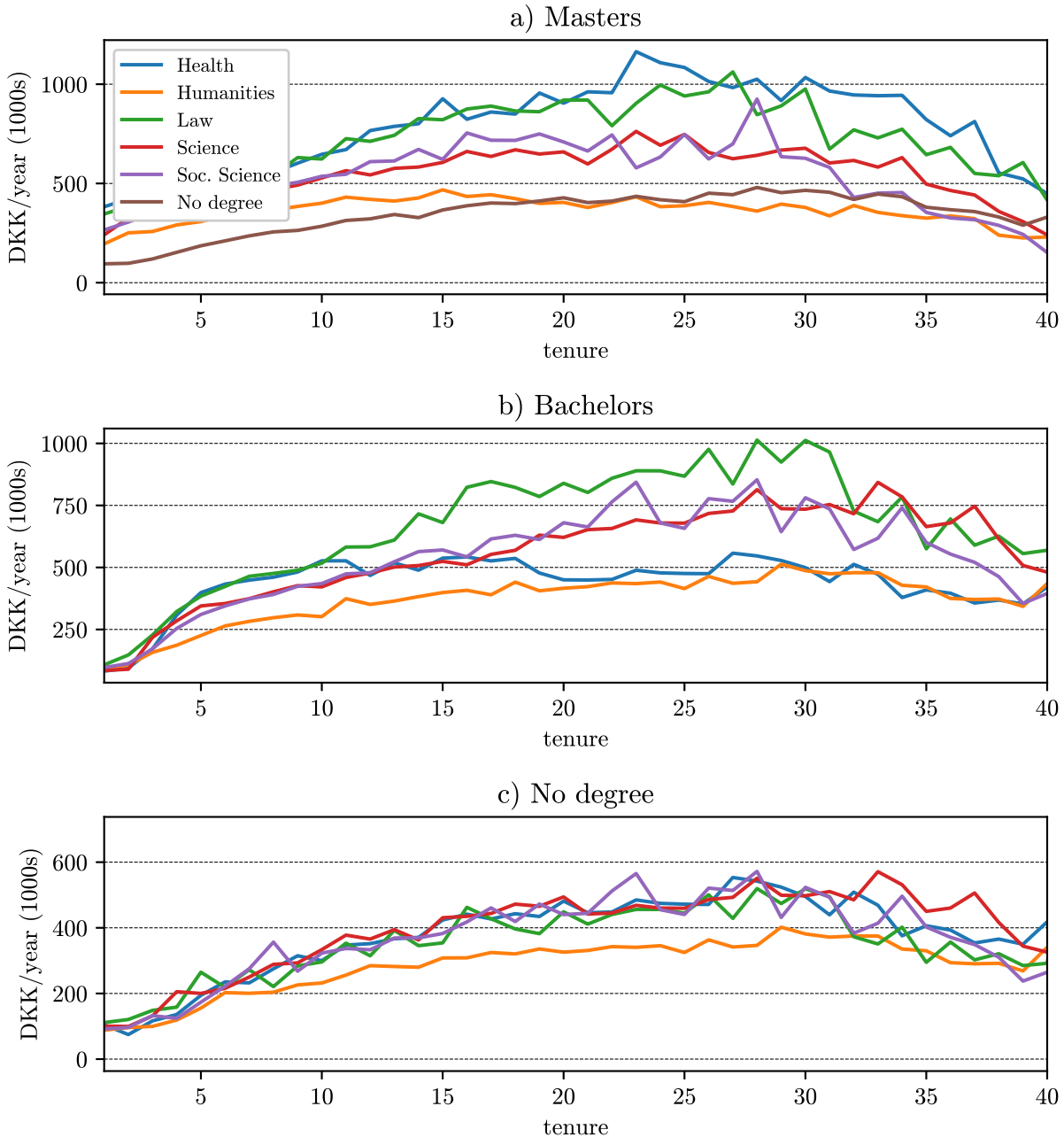
| | Social Science (n=4,345) | | Law (n=3,621) | |
|---|---|---|---|---|
| | Mean | SD | Mean | SD |
| *Individual-level:* | | | | |
| Female | 0.58 | 0.49 | 0.61 | 0.49 |
| Age at entry | 21.17 | 3.25 | 21.32 | 4.00 |
| High school GPA | 9.69 | 1.52 | 8.49 | 1.50 |
| *Semester-level:* | | | | |
| Registered credits | 24.12 | 12.61 | 25.74 | 12.35 |
| Passed credits | 21.99 | 12.62 | 22.46 | 12.85 |
| Semester GPA | 8.33 | 2.57 | 7.52 | 2.58 |
| Hours per semester | 216.27 | 229.76 | 266.38 | 247.01 |
| Hourly wage | 122.74 | 104.03 | 129.31 | 89.55 |

Note: Semester GPA is the weighted average of grades a student receive in a semester. In about 25 pct of semesters, the student does not pass any course or receive only a "passed" mark. Semester GPA is missing in these cases.

Social science students enter with a higher high school GPA and also receive higher grades at university. Law students register and pass more credits, and they work more hours at a higher wage.

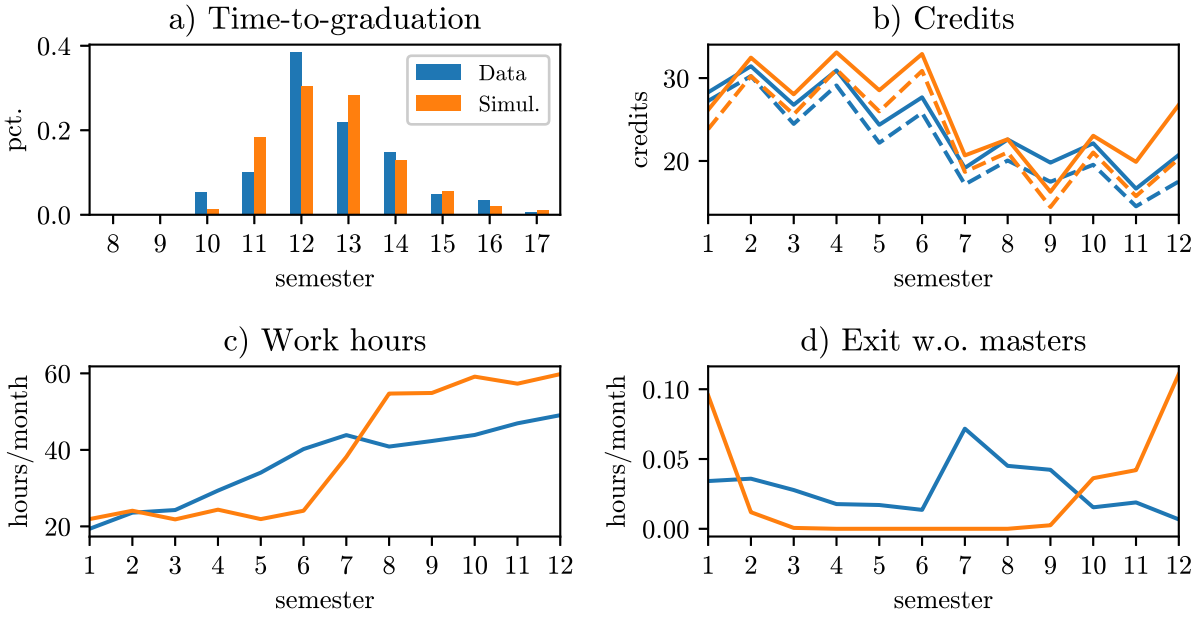
Figure A.3 shows the data and model simulation for the Social Science faculty separate. The model fit is similar in quality to the Faculty of Law fit. Again, the timing of the dropout is different in the model fit. Social Science students have a peak in dropout in the 7th semester after attaining a bachelor's degree. In the model, they stay enrolled longer until they attain a bachelor's degree and choose to exit.

Figure A.2: Income Profiles by Faculties




Note: First-time UCPH graduates and dropouts from earlier cohorts in Statistics Denmark's registers. Students might have attained degrees later. Wage and business income measured in 2015. Averages by each tenure year, calculated as 2015 minus year of graduation. There have only been bachelor's graduates (used to be combined programs) since 1994. Hence, average income after 21 years of tenure is extrapolated with growth rates from master's graduates.

Figure A.3: Model Fit - Social Science




Note: Simulation with model parameters of students at the Faculty of Law. b) Solid lines are registered credits. Dashed are passed credits.

## A.4 Ordered Logits for Passed Credits

In the estimation $p_0, ..., p_j, ..., p_{c^*}$ are the passing probabilities for each credit level,$e$, equal to or below the registered level $c^*$. $p_0$ is the probability of passing zero credits. A latent index variable is given with $u$ being standard logistic:

$$y^* = x'\xi + u \tag{15}$$

The passed credit level is then given by the latent index and cutoff values, $\lambda$:

$$e = j \text{ if } \lambda_{j-1} < y^* \leq \lambda_j \tag{16}$$

Where $\lambda_{-1} = -\infty$ and $\lambda_{c^*} = \infty$. Passing probabilities of each level is given by:

$$p_j = F(\lambda_j - x'\xi) - F(\lambda_{j-1} - x'\xi) \tag{17}$$

Since we have seven possible levels for $c^*$ we estimate six ordered logits (the zero credit choice is trivial). We estimate the $\xi$ and cut-off $\lambda$'s by maximum likelihood. For each level of $c^*$ passing probabilities are predicted for each individual by:

$$\hat{p}_j = F(\hat{\lambda}_j - x'\hat{\xi}) - F(\hat{\lambda}_{j-1} - x'\hat{\xi}) \tag{18}$$

Table A.3: Passed Credits

| | 7.5 ECTS | | 15 ECTS | | 22.5 ECTS | | 30 ECTS | | 37.5 ECTS | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Est. | SE | Est. | SE | Est. | SE | Est. | SE | Est. | SE |
| Female | 0.394 | 0.071 | 0.417 | 0.050 | 0.469 | 0.040 | 0.377 | 0.022 | 0.326 | 0.036 |
| High school GPA | 0.017 | 0.022 | 0.045 | 0.015 | 0.062 | 0.013 | 0.134 | 0.007 | 0.091 | 0.011 |
| $GPA_{t-1}$ | 0.364 | 0.023 | 0.330 | 0.015 | 0.373 | 0.013 | 0.333 | 0.007 | 0.288 | 0.010 |
| Work hours (100s) | -0.009 | 0.014 | -0.028 | 0.011 | -0.031 | 0.010 | -0.033 | 0.006 | -0.020 | 0.009 |
| Spring | -0.094 | 0.072 | 0.010 | 0.052 | 0.333 | 0.041 | 0.527 | 0.023 | 0.791 | 0.039 |
| Humanities | -1.348 | 0.145 | -0.099 | 0.102 | -0.602 | 0.085 | -0.102 | 0.043 | -0.625 | 0.065 |
| Law | -0.297 | 0.130 | -0.222 | 0.124 | 0.090 | 0.076 | 0.143 | 0.045 | 0.306 | 0.067 |
| Science | -0.785 | 0.130 | -0.588 | 0.102 | -0.553 | 0.075 | -0.375 | 0.041 | -0.365 | 0.062 |
| Social Science | -0.410 | 0.143 | -0.244 | 0.116 | 0.158 | 0.079 | -0.036 | 0.048 | -0.018 | 0.068 |
| $\lambda_1$ | -1.125 | 0.183 | 0.709 | 0.138 | 0.537 | 0.111 | 0.147 | 0.066 | -0.377 | 0.104 |
| $\lambda_2$ | | | 1.315 | 0.138 | 1.563 | 0.110 | 0.671 | 0.064 | 0.230 | 0.100 |
| $\lambda_3$ | | | | | 2.200 | 0.111 | 1.400 | 0.064 | 0.915 | 0.098 |
| $\lambda_4$ | | | | | | | 2.150 | 0.064 | 1.673 | 0.098 |
| $\lambda_5$ | | | | | | | | | 2.579 | 0.100 |
| N | 4,838 | | 10,658 | | 14,851 | | 63,726 | | 14,906 | |

Note: Ordered logistic regressions. The unit of observation is semesters. Work hours are within the same semester as the course registrations.

See Table A.3 for estimation results.

## A.5 Study Subsidy Limit

There is a limit to how much income students are allowed to have while receiving the study subsidy (SU). If yearly income is above DKK 148,500 the excess income is deducted in the subsidy at the end of the year. In the model, we ignore this limit as it is only binding for a few students, and we do not see clear bunching close to the limit in Figure A.4.

## A.6 Choice Probabilities

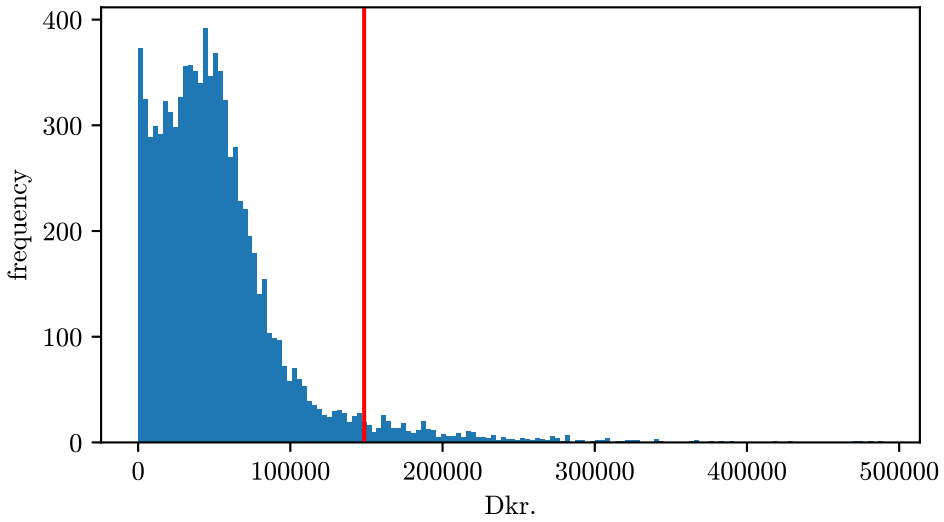
By assuming $\epsilon$ in section 3.1.1 is iid extreme value type I[23] with location parameter $\mu = 0$ and scale parameter $\delta$ we get a very simple expression for the *integrated* value function:

$$IV(s) = \delta \cdot \left( \gamma + \log \left[ \sum_{j=1}^{J} \exp \left( \frac{v_a(s)}{\delta} \right) \right] \right) \tag{19}$$

In this setup, all choices enter equally in the stochastic dimension. This has the benefit that it's very simple to solve *and* estimate. It has the disadvantage that the similar choices do not have a positive correlation between alternatives. For all choice pairs $(i, j)$ we have that

[23] CDF: $G(\epsilon_a) = \exp\left(-\exp\left(-\frac{-\epsilon_a}{\sigma}\right)\right)$

Figure A.4: Study Subsidy Limit




Note: Income distribution for UCPH students in 2015.

$corr(\epsilon_i, \epsilon_j) = 0$. This can sometimes be a very restrictive assumption. If we want to allow for $corr(\epsilon_i, \epsilon_j) \neq 0$ we cannot define the stochastic nature of the shocks individually. Instead, we specify a multivariate distribution for the shock *vector* according to

$$G(\epsilon) = \exp\left[-\sum_{r=1}^{R}\left(\sum_{a\in P_r}\exp(-\frac{\epsilon_a}{\sigma_r})\right)^{\frac{\sigma_r}{\delta}}\right] \tag{20}$$

and get the nested logit model. In the expression, we introduce partitions $P_r$, of which there are $R$. These are the nest specific choice sets. Each partition has a parameter $\delta > \sigma_r > 0$, which is a measure of the degree of independence in unobserved utility among the alternatives in nest $r$. The integrated value function looks slightly different under this distributional assumption

$$IV_n(s) = \delta \cdot \left(\gamma + \log\left[\sum_{r=1}^{R}\left(\sum_{a\in P_r}\exp\left(\frac{v_a}{\sigma_r}\right)\right)^{\frac{\sigma_r}{\delta}}\right]\right) \tag{21}$$

In our case, we have two partitions, where the last $P_2$, has just one element. Then we get

$$\mathbf{IV}(s) = \delta \cdot \left( \gamma + \log \left[ \sum_{a \in P_1} \exp\left( \frac{v_a}{\sigma_1} \right)^{\frac{\sigma_1}{\delta}} + \sum_{a \in P_2} \exp\left( \frac{v_a}{\sigma_2} \right)^{\frac{\sigma_2}{\delta}} \right] \right) \tag{22}$$

$$= \delta \cdot \left( \gamma + \log \left[ \sum_{a \in P_1} \exp\left( \frac{v_a}{\sigma_1} \right)^{\frac{\sigma_1}{\delta}} + \exp\left( \frac{v_{P_2}}{\sigma_2} \right)^{\frac{\sigma_2}{\delta}} \right] \right) \tag{23}$$

$$= \delta \cdot \left( \gamma + \log \left[ \sum_{a \in P_1} \exp\left( \frac{v_a}{\sigma_1} \right)^{\frac{\sigma_1}{\delta}} + \exp\left( \frac{v_{P_2}}{\delta} \right) \right] \right) \tag{24}$$

Setting $\delta = 1$ and letting the with-in first partition correlation go to zero (by letting $\sigma_1 \to \delta$) we get:

$$\lim_{\sigma_1 \to \delta} \mathbf{IV}(s) = IV(s) \tag{25}$$

of the extreme value type I assumption.

Under the nested logit assumptions, we can write up the conditional choice probabilities in closed form. The probability of choosing a specific choice is generally the product of the probability of choosing the choice within its nest times the probability of choosing the nest.

$$P(\text{exit}|s) = 1 \times \frac{\exp(v(\text{exit}, s))}{\exp(v(\text{exit}, s)) + \left[ \sum_{j \in A_{\text{study}}} \left( \frac{\exp(v(c_j^*, w_j, s))}{\sigma} \right) \right]^{\sigma/\delta}} \tag{26}$$

$$P(c^*, w|s) = \frac{\exp\left( \frac{v(c^*, w, s)}{\sigma} \right)}{\sum_{j \in A_{\text{study}}} \exp\left( \frac{v(c_j^*, w_j, s)}{\sigma} \right)} \times (1 - P(\text{exit}|s)) \tag{27}$$

where we use $\sigma$ to denote the only nest's weight, since the exit weight disappears from all expressions.

# Chapter 3
# Equilibrium Conditions and Solution Methods for Directional Dynamic Oligopoly Games

# Equilibrium Conditions and Solution Methods for Directional Dynamic Oligopoly Games

Patrick Kofod Mogensen

**Abstract**

In this paper, I derive equilibrium conditions for sub-stages in directional dynamic games with different model specifications in terms of number of actions, number of players, and exogenous (non-)directional states. I show how to use these to solve for all Markov Perfect Equilibria using Recursive Lexicographical Search. I add to the existing literature by deriving the needed equilibrium conditions needed to solve these games, and provide details on how to solve the sub-stages. I show how to solve the system of multivariate polynomial equations in complete information games using all-solution methods and propose a way to solve the more complex system of equations using interval arithmetic in incomplete information versions of some of the games. Full solution methods are important if the aim is to characterize the potential market configurations that can obtain, or if the goal is to estimate structural parameters in a model of dynamic, strategic interaction.

## 1 Introduction

Solving and estimating dynamic games is an active and important field of research. For example, most cases in industrial organization require some form of strategic interaction to be realistic. Estimation is possible to some extent, but the presence of multiple equilibria causes problems. Even if the only objective is to solve and analyze a given model, it's required to have robust tools, such that all possible types of equilibria are found with certainty. For an empirically relevant type of model called directional, dynamic games (DDGs) Iskhakov et al. (2015) introduced a method for using the structure in the models to search for all equilibria. If the so-called sub-stages which are relatively small problems embedded in the overall problem can be solved, then the full model can also be solved systematically.

In this paper, I extend the analysis of directional, dynamic games (DDGs) by deriving equilibrium conditions for multinomial, multiplayer, and exogenous non-directional state

DDGs. Expressions that enables the solution of all sub-stages are derived, and I provide all the steps necessary to solve a complete and incomplete information version of the extensions. In Iskhakov et al. (2015), Iskhakov et al. (2018), Iskhakov et al. (2020) they provide some approaches to solving their sub-stages. Their approach is details and driven by the structure, but may also be hard to implement correctly for many researchers due to the combination of several techniques and methods. I suggest to use two general-purpose root-finding methods that are appropriate to find the solutions of the sub-stages.

The point of departure for this paper is Iskhakov et al. (2015) who developed a full solution method for DDGs. There are two requirements: the states must be directional, and it must be possible to solve what they call sub-stages of the over-all game. This opens up to completely solving such games to study different market structures and can also be helpful in NFXP (Rust (1987); John (1988); Rust et al. (1994); Iskhakov et al. (2020)) style estimation procedures. They provide a specific example of such a game and show how each sub-stage can be solved using nothing more than a few derivations of corner cases mixed with solving second-order polynomials. They can solve their complete information sub-stages using the formula for second-order polynomial roots, and they use a second-order best response solution method for their incomplete information sub-stages. Since the equilibrium conditions in this paper can involve higher-order, multivariate polynomials I contribute to thntere RLS literature by suggesting a different strategy based on homotopy methods, and I propose a less convoluted method of solving the incomplete information sub-stage equations.

Neither this paper nor Iskhakov et al. (2015) is the first to recognize the polynomial, or near-polynomial, structure in games. Two important contributions to applying full-solution methods for games whose equilibrium conditions reduce to polynomial equations are Datta (2010) and Judd et al. (2012). The former suggests using methods based on Gröbner bases and the latter suggests all-solution homotopy continuation methods. This paper places itself in that literature by using similar methods and software to suggest an approachable way to solve sub-stages in RLS. Iskhakov et al. (2015, 2018) use a second-order best reponse method that works for their binary choice duopoly game with incomplete information and their complete information games reduce to second-order polynomialsi. I suggest the use of interval arithmetic to solve the equations in the incomplete information case where the conditions are not systems of multivariate polynomials, and either interval methods or all-solution homotopy continuation methods in the complete information case.[1]

This paper proceeds as follows. In section 2 the introduce the model class, in section 3 I present the equilibrium conditions and discuss the steps needed to solve the complete and

[1] An earlier version of this paper used Gröbner bases but without the need for parametric systems, the cost of symbolic operations seems to be prohibitive for complicated models with high order polynomial systems.

incomplete versions of the game, in section 4 I discuss the numerical tools I will use to solve the sub-stages, in section 5 I find equilibria in some example models, and finally section 6 concludes.

# 2 A Class of Dynamic Directional Games

In this section, I present the class of games I characterize equilibrium conditions for. The notation and assumptions follows descriptions like Rust et al. (1994); Aguirregabiria and Mira (2010) for single agent models.

**Assumption 1 (Finite players)** *There is a finite number of players indexed* $1, 2, \ldots, \mathcal{N}$.

The assumption is practical. As will be shown, there is at least one equation per player in the equilibrium conditions. Solving an infinite-dimensional system of equations might have theoretical importance, but it is only possible in special cases. Do note, that it is not necessarily a given in all games that the number of players is finite. For example, in the platform game in Dubé et al. (2010) there is a continuum of agents in a game that otherwise fits the setup in this paper.

**Assumption 2 (Partially Directional State Space)** *The state space,* $\mathcal{S}$*, can be partitioned as a directional part* $\mathcal{D}$*, a non-directional part* $\mathcal{X}$*, and the shocks* $\mathcal{E}(d, x)$*. Lower case letters* $s$*,* $d$*,* $x$*, and* $\epsilon$ *denote elements of* $\mathcal{S}$*,* $\mathcal{D}$*,* $\mathcal{X}$*, and* $\mathcal{E}$*.* $\mathcal{D}$ *and* $\mathcal{X}$ *are discrete and finite, and* $\mathcal{X}$ *is allowed to be empty, and* $\mathcal{E}(d, x) \subseteq \mathcal{E}_1(d, x) \times \ldots \times \mathcal{E}_{\mathcal{N}}(d, x)$.

The directional part of the state space is essential to the approach in Iskhakov et al. (2015) and essentially means that those states can only move in one direction. This is essential to the recursive strategy.

**Assumption 3 (Incomplete Information)** *Player* $i$ *observe their shocks,* $\epsilon_i \in \mathcal{E}_i$*, to choice specific payoffs, but only know the distribution of the other players' shocks* $\{\epsilon_j\}_{j \neq i}$*. All directional,* $d \in \mathcal{D}$*, and non-directional states,* $x \in \mathcal{X}$*, are known to all players.*

**Assumption 4 (Complete Information)** *All players observe all shocks,* $\{\epsilon_i\}_{i=1}^{\mathcal{N}}$ *where* $\epsilon_i \in \mathcal{E}_i$*. All directional,* $d \in \mathcal{D}$*, and non-directional states,* $x \in \mathcal{X}$*, are known to all players.*

The two previous assumptions are mutually exclusive. When I mention incomplete information games here and below I refer to models where there are shocks present in the model. Complete information games are games where the shock space is the empty set.

**Assumption 5 (Discrete and finite actions)** *The action space for each player in each state, $\mathcal{A}_i(d,x)$, is discrete, and finite, and the $J_i(d,x)$ elements in each action set are enumerated $1,\ldots,J_i(d,x)$.*

Notice, that the number of choices is allowed to depend on the state the agents are in. Actions affect expected payoffs and transition probabilities from current $(d,x)$ to future $(d',x')$. They affect payoffs through state transitions within the period, shock realizations and economic consequences of actions such as investment costs, fixed costs, and revenue. To describe the strategic behavior of the agents I introduce the stationary Markov behavior strategies or just strategy for short[2]

**Definition 1 (Strategies)** *A strategy $\sigma_i(s)$ is represented by a vector*

$$\left(p_i^1(s), p_i^2(s), \ldots, p_i^{J_i(d,x)}(s)\right)$$

*that defines a categorical distribution. Under strategy $\sigma_i(s)$ agent $i$ plays action $j$ with probability $p_i^j(s)$ and only depends on the current state $s$.*

Given the environment described by the states each player seeks to maximize expected, discounted profits, as seen from the initial period. We will assume an infite horizon and then the solution to the utility maximization problem for player $i$ is given by the using Bellman's principle of optimality that gives us the recursive expression

$$V_i(s) = \max_{a_i \in \mathcal{A}_i(s)} \left\{ U_i(d,x,a_i \,|\, \sigma_{-i}(s)) + \epsilon_i(a_i) + \beta \int V_i(s_{t+1}) \mathrm{d}F_i(s'|s,a_i,\sigma_{-i}(s))) \right\}, \quad (1)$$

where $\mathbf{a}_i$ is the optimal policy or best response for player $i$ to the environment and the belief about other players strategies,. Here, $U(d,x,a_i \,|\, \sigma_{-i}(s))$ is the expected instantaneous utility because each player is uncertain about the actions of the other players and potentially also the immediate outcomes of their own actions. Agent $i$ will generally not know the actions of the other players and their own actions might only control their environment in a stochastic way. For example, if the choice is to build a new factory this period they might not be guaranteed to get a permit. If they don't get a permit they won't have building costs this period. This means that solving the games amounts to solving the Bellman equations for all players at once. I will assume that the shock space $\mathcal{E}$ is continuous which is why the expectation is expressed as an integral. In the single agent literature the Bellman equation is a contraction mapping. When seen as a system of equations the mapping that evaluates the system for some candidate value function vector will generally not be a contraction.

[2]See (Doraszelski and Escobar, 2010, page 8) or (Iskhakov et al., 2015, page 7))

This framework puts a restriction on the types of equilibria and defines what we mean by a best-response. A best-response is a solution conditional on the strategies of the other players to the problem in eq. (1). We return to the equilibrium concept below.

**Assumption 6 (Conditional Independence)** *The transistion probability function, $\pi_S$, denoting the probabilities of $s$ transitioning to a state $s'$ next period satiesfies a conditional independence property*

$$\pi_i^{\mathcal{S}}(s'|s,a) = \pi_i^{\mathcal{D},\mathcal{X}}(d',x'|d,x,a,\sigma_{-i})\pi_{\mathcal{E}}(e'|e,a). \tag{2}$$

The conditonal independence assumption allows us to separately evaluate the integral for the choice specific shocks $\epsilon$ and the other states. Further simplification comes from the following assumption.

**Assumption 7 (Logit)** *The shocks are iid, across players, time and choices, extreme value type I distributed with adjusted scale parameter $\eta$ and a mean of 0.*[3] *This implies that $\mathcal{E}_i(d,x) \subseteq \mathbb{R}^{J_i(d,x)}$.*

The power of this assumption with the additive structure of the shocks above is that the integrated value function for each player

$$IV_i(d,x) \equiv \int V_i(d,x,\epsilon_i)\ \mathrm{d}G(\epsilon_i) \tag{3}$$

fully characterizes the solution of the original model given the assumptions of $G$ which the cummulative density function of the shocks and that I can find a simple expression for it. The function can be found as the unique solution to the integrated Bellman equations that can be shown to have the simple form of

$$IV_i(d,x) = \eta \log\left(\sum_{j=1}^{J_i(d,x)} \exp\left(\frac{v_i^j(d,x)}{\eta}\right)\right) \tag{4}$$

where the choice specific value functions $v_i^j$ are defined as

$$v_i^j(d,x) = U_i(d,x,a_i \,|\, \sigma_{-j}(s)) + \beta_i \sum_{d'\in\mathcal{D}} \sum_{x'\in\mathcal{X}} IV_i(d',x')\pi_i^{\mathcal{D},\mathcal{X}}(d',x'|d,x,a_i,\sigma_{-i}) \tag{5}$$

[3] In other words, the location parameter is $-\sigma\gamma$ with $\gamma$ being the Euler-Mascheroni constant. This means that the *mode* is not zero, but for this distribution it's impossible to have both. If a location parameter of 0 is preferred, then the integrated value function simply changes to $IV(d,x) = \sigma(\gamma + \log\sum \exp u)$. Since the mean is now positive, the specific solutions might shift slightly, depending on the scale parameter used.

Notice, that not all $d' \in \mathcal{D}$ will be feasible from all current $d$, but this is taken care off by assigning zero probability to such transitions. This gives a very simple expression for the conditional choice probabilities

$$p_i^k(x) = Pr_i(a = k|x) = \frac{\exp\left(\frac{v_i^k(d,x)}{\eta}\right)}{\sum_{l=1}^{J_i(d,x)} \exp\left(\frac{v_i^l(d,x)}{\eta}\right)}. \tag{6}$$

I now return to define what equilibrium concept I'm going to use. The models I will present fit in to the RLS framework. The authors define their equilbrium concept in Iskhakov et al. (2015)[Definition 1] to be stationary Markov Perfect Equilibria. More further discussion, see Doraszelski and Pakes (2007), Doraszelski and Escobar (2010) on MPEs in dynamic industrial organization.

**Definition 2** *Markov Perfect Equilibrum A Markov Perfect Equilibrium is a set of integrated value functions* $\{IV_i(d,x)\}_{i=1}^N$ *also defined for all states of the game that solve Equation 4 as well as the optimal policies given by the set of strategy functions* $\{\sigma_i(s)\}_{i=1}^N$ *with consistent beliefs for all players for all* $s \in \mathcal{S}$.

I have now described most of the requirements I will impose in the derivation of the equilibrium conditions in this paper. Since I am going to be solving the equilibrium conditions in the choice probability space, it is useful to write the Bellman equation slightly differently.

**Assumption 8 (Expected utility)** *The expected utility is assumed to be of the following form*

$$U_i(d,x,a_i|\sigma_{-i}) = \sum_{\alpha \in \bar{\mathcal{A}}(d)} \left[ \left( \prod_{j \neq i} \sigma_j(d,x) \right) u_i(d,x,a) \right] \tag{7}$$

*where* $\bar{\mathcal{A}}(d) \equiv \mathcal{A}_1(d) \times \ldots \times \mathcal{A}_{i-1}(d) \times \{a_i\} \times \mathcal{A}_{i+1}(d) \times \ldots \times \mathcal{A}_{\mathcal{N}}(d)$ *such that* $a$ *is a realized action for all players.*

since I pass in the realized actions $a$ into $u_i(d,x,a)$ there is still room to model $u_i(d,x,a)$ as expected utility given the actions. This expectation then represents something like stochastic costs given a choice. Since $x$'s are exogenous, we can write

$$\pi^{\mathcal{D},\mathcal{X}}(d',x'|d,x,a_i,\sigma_{-i}) = \pi^{\mathcal{D}}(d'|d,a_i,\sigma_{-i})\pi^{\mathcal{X}}(x'|x) \tag{8}$$

which totally decouples the transition probabilities.[4] Since I'm looking at the transition probabilities as seen from agent $i$ the effect of the actions of the other players are only valid

[4]Given the derivations in the rest of the paper it is clear that it is possible to have a more complicated structure here. Maybe some exogenous states are only present in some directional states such that $\pi^{\mathcal{X}}(x'|x,d)$ may be more appropriate, but we don't do that here.

up to the belief over other players' strategies. To that end, I introduce $\pi^{\mathcal{A}}$ that defines the probability of $d$ transitioning into $d'$ given all the actions, not strategies.

$$\pi^{\mathcal{D}}(d'|d, a_i, \sigma_{-i}) = \sum_{a \in \bar{\mathcal{A}}} \left( \prod_{k \neq i} \sigma_k^{a_k}(d, x) \pi^{\mathcal{A}}(d'|d, a) \right) \tag{9}$$

or in words: seen from player $i$'s point of view, the transition probability given their belief over the other players strategies is found by evaluating the sum of $(d'|d)$-transition probabilities given the complete action vector $a$ weighted by probability of each action occuring given the belief of $\sigma_{-i}$. Then, I can write our choice specific value functions as

$$v_i^j(d, x) = \sum_{d' \in \mathcal{D}} \sum_{a \in \bar{\mathcal{A}}} \left( \prod_{k \neq i} \sigma_k^{a_k}(d, x) \pi^{\mathcal{A}}(d'|d, a) \right) \left[ u_i(d, x, a) + \beta_i \sum_{x' \in \mathcal{X}} IV_i(d', x') \pi^{\mathcal{X}}(x'|x) \right] \tag{10}$$

This formulation is useful when $\pi^{\mathcal{A}}(d'|d, a)$ is degenerate in the sense that a pair $(d, a)$ uniquely and deterministically determines $d'$.

## 2.1 Structure of the directional space

As mentioned above, the RLS theory and algorithm were laid out in Iskhakov et al. (2015) who explain in great detail how to implement and devise the overall iterative scheme that ensures that sub-stage equilibriums are efficiently and exhaustively checked in ways they can produce the equilibria for dynamic directional games. To do this they introduce equilibrium selection strings, variable metric arithmetic, and many more concepts that make it possible to write up a program the solves the full game. However, the most important part for this paper is the fact that if you recurse in the opposite direction of the directionality of the game, you can partition the game into sub-stages ordered according to the progression of the directional state variables. Within a sub-stage time is allowed to advance but all directional stages are kept constant. The procedure is a multivariate version of backwards induction because there can be more than one directional state. Typically, there will at least be one directional state variable per agent. Solving *from the back* like this means that I can keep track of the effect of choosing different equilibria in different sub-stages, and it also means that all value functions I need to solve a sub-stage, besides the value functions for the state(s) in that sub-stage, are known. This gives enough structure to derive the equilibrium conditions I will show below. In subsection 2.2 I provide an example of how the stages can be structured that will provide an intuitive understanding of what a sub-stage is.

To fix ideas, I look at games where the state transitions have a particular form given the actions. The directional part of the state space, $\mathcal{D}$, can be seen as a vector of state indices

for each player:

$$d = (d_1, d_2, \ldots, d_i, \ldots, d_{\mathcal{N}})$$

If the actual choices of each player is given as $a = (a_1, a_2, \ldots, a_{\mathcal{N}})$ then player $i$ has to use the following transition probabilities when taking the strategies of the other players into account

$$\pi_i(d', x'|d, x, a_i, \sigma_{-i}) = \pi_i^{\mathcal{D}}(d'|d, a_i, \sigma_{-i}) \cdot \pi^{\mathcal{X}}(x'|d, x) \tag{11}$$

$$\pi_i^{\mathcal{D}}(d'|d, a_i, \sigma_{-i}) = \prod_{n=1}^{\mathcal{N}} \pi_i^{\mathcal{D}}(d'_n|d, a_i, \sigma_{-i}) \tag{12}$$

$$\pi_i^{\mathcal{D}}(d_j + k - 1|d, a_i, \sigma_{-i}) = \sigma_j^k(d), \quad j \neq i \tag{13}$$

$$\pi_i^{\mathcal{D}}(d_i + k - 1|d, a_i = k, \sigma_{-i}) = 1 \tag{14}$$

$$\pi_i^{\mathcal{D}}(d_i + k - 1|d, a_i \neq k, \sigma_{-i}) = 0 \tag{15}$$

Such that the choice *index* indicates the increments of the state values. These values can be thought of as indices to a categorical state or an actual count variable. The transitions are then deterministically determined by each player's choice. Any player $i$ knows that player $j$ directly controls $d'_j$ through their actions, but they only have beliefs of their behavior strategies. They can only determine $d'_i$ with certainty. This means that $d' = d + a - 1$ such that I get

$$v_i^j(d, x) = \sum_{a \in \bar{\mathcal{A}}} \left( \prod_{j \neq i} \sigma_j(d, x) \right) \left[ u_i(d, x, a) + \beta_i \sum_{x' \in \mathcal{X}} IV_i(d + a - 1, x') \pi^{\mathcal{X}}(x'|x) \right] \tag{16}$$

$$= \sum_{a \in \bar{\mathcal{A}}} \left( \prod_{j \neq i} \sigma_j(d, x) \right) \tau_i(d, x, a) \tag{17}$$

$$\tau_i(d, x, a) = \left[ u_i(d, x, a) + \beta_i \sum_{x' \in \mathcal{X}} IV_i(d + a - 1, x') \pi^{\mathcal{X}}(x'|x) \right] \tag{18}$$

where $\tau_i$ is the choice specific value function taking all actions as given, not just that of player $i$.

For a two player game with binary choice and no non-directional states, this reduces to

$$v_1^j(d, x) = \sigma_2^1(d)\tau_1(d, x, (j, 1)) + \sigma_2^2(d)\tau_1(d, x, (j, 2))$$
$$v_2^j(d, x) = \sigma_1^1(d)\tau_2(d, x, (1, j)) + \sigma_1^2(d)\tau_2(d, x, (2, j)).$$

This form is not essential for the derivation method. If different rules were given some of the summations involved in expressing the expectations over the possible future states would become more complicated, but the form of the resulting equilibrium conditions would keep its overall properties. To avoid clutter in the main text I stick to transition structure from

Equation 15 above because it fits the example model, but I provide some details for the more general model in Appendix F. This extended model would handle the uncertain investments for example, and could be applied if zoning laws made it uncertain if the fast food chains in this paper's example section would be sure whether a decision to open a new restaurant would materialize or not. Though, it only works if there is at least one choice that leads to a certain movement from the current directional state to a different one.

In fig. 1 I show some different types of restrictions on the geometry of the directional state space. If there are two firms with an endogenous state associated with each, and there is simply an upper limit to this state's value, then the state space is as in the first panel. This is the case in our expansion model I will present below. In the second panel, there is a model where the restriction is put on the collection of states. For example, if a market can only support five fast-food restaurants, but each restaurant would be able to open ten each, the state space would look like this. It could also be a mix, as shown in the third panel. Here, each firm can support four restaurants, but there can be no more than six restaurants in total for institutional reasons. Depending on the specific game these restrictions can impact the presence of multiple equilibria. We also clearly see how the directional structure leads to the possibility of dividing the game into sub-stages represented by each dot. Once a state represented as a dot has been solved for, I can solve the dot to the west and south. Then with those three solutions in hand, I can solve the south-west dot.

The point of showing these is to identify some common cases. Of course, there may be models that deviate from this pattern, but a lot of the structure from the exposition here is re-useable. The following enumeration of the different types of states borrows from Iskhakov et al. (2015). Whenever neither player can advance their endogenous state by taking an action, I call it a corner state. These are colored blue in fig. 1.

**Definition 3 (Corner state)** *A corner state is a state where all endogenous directional states are in their absorbing state.*

It could just as well be decreasing depending on the application. Another kind of state is the *edge* state. These were red or orange in fig. 1.

**Definition 4 (Edge state)** *An edge state is a state where at least one endogenous, directional state is absorbing, and at least one endogenous, directional state is not.*

In fig. 1 only the first and last panel had both of the types of states introduced so far. The second does not have any edge states. The last type of state I will explicitly denote is the interior state. In fig. 1 these are shown as black dots.

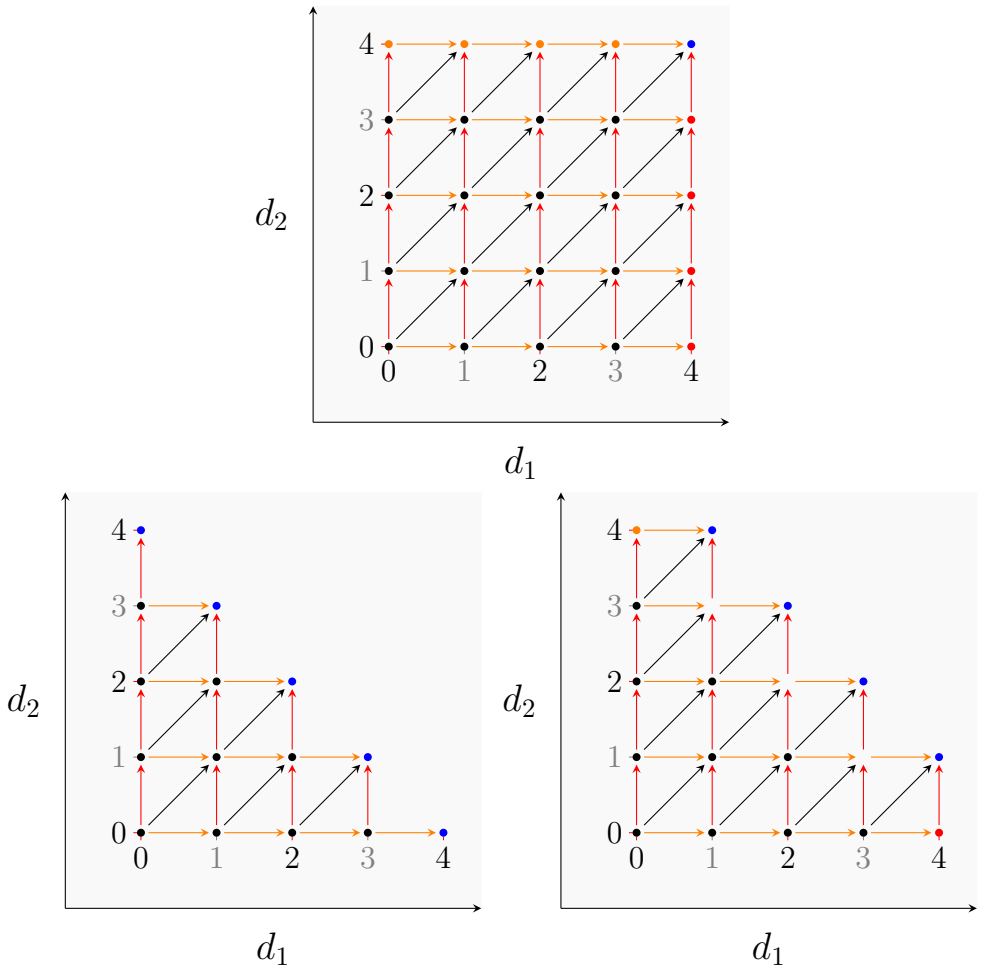



Figure 1: Three posibilites (left to right): cap on each state, cap on sum of states, and a mix of the two.

**Definition 5 (Interior state)** *An interior state is a state where at least two directional states are not in their absorbing state.*

This means that there is still room for movement in more than one directional state. Unless the actions of the player have consequences to the other players beyond changing states, the interior states are those that can have actual strategic behavior and multiple equilibria. This is because several players can affect state transitions. If only one player can affect state transitions this is the usual single-agent situation. If there are more than two players in total this could be the case where one player is in an absorbing state, but the other two are still actively trying to affect their situation.

## 2.2 Expansion Game

We now present a model of market expansion where two firms operate in a market and have to choose to open a new store or not each period. The model is heavily inspired by Aguirregabiria et al. (2009) which builds on Toivanen et al. (2005). To begin with, I ignore the macro market state dynamics, but later I allow for the inclusion of exogenous states. It can easily be extended to multiple openings per period and multiple agents, as will shown. Time is discrete, and there is a maximum number of stores each firm wants to have, or are allowed to have, in a given market, $\bar{d}_j$ for $j \in 1, 2$. The directional state space is then

$$\mathcal{D} = \{(d_1, d_2) \in \mathcal{Z}^2 \mid 0 \leq d_j \leq \bar{d}_j,\ j \in \{1, 2\}\} \tag{19}$$

The feasible state transitions are illustrated in fig. 2. All orange arrows are state evolutions where only firm 1 opens a store, the black arrows indicate co-movement, and the red arrows represent only firm 2 opening a store. The purple dot is the absorbing state, the orange dots are where only firm 2 is in its absorbing state, and similar for the red dots and firm 1. The black dots are states where both firms can expand. It is impossible to close a store in this model. Toivanen et al. (2005) motivate this by the fact that exits a very rare phenomenon in their data.

We will now describe how the stages are ordered. The key to the approach is to start from the back. Much like regular backwards induction you first look at the last interaction, then the second last, and so on back to the very first interaction in the game. In this model, the recursion starts at the absorbing state $(4, 4)$ with $D = 4 + 4 = 8$ stores. The previous stage is where $D = 4 + 4 - 1 = 7$, which is the red dot just below, $(4, 3)$, and the orange dot to the left, $(3, 4)$, of the absorbing state. The third last stage consists of dots on the line connecting $(4, 2)$ and $(2, 4)$, and so we continue all the way back to $(0, 0)$. I will come back to this when solving the incomplete information game but I'm calling this an expansion

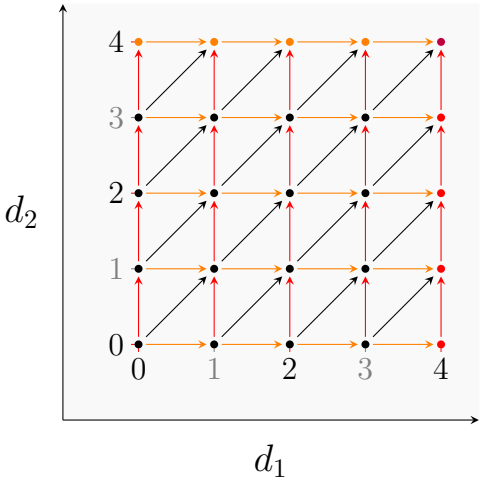



Figure 2: Illustration of $\mathcal{D}$ in the expansion game.

game because the interesting part of the dynamics is the evolution of the directional state: the number restaurants. In what might be called an *entry* game, the more interesting thing would be whether an agent was present with a firm in a market or not. This would reduce our state space significantly, but in such a model directionality might not be the best modeling choice, because entry models are often entry/exit models. In such a model, alternative costs are very important. If an agent doesn't enter a fast food market, they might invest their money elsewhere and get an expected return. In the model presented here, the reference profits are simply zero, so the incentive to move out of states with zero elements will be very strong.

To have a chance of solving this model, we have to describe the fundamentals of the model. The firms discount the future according to the discount factors $\beta_1$ and $\beta_2$ respectively. In each state of each stage, the two firms engage in static Cournot competition. Assume that the marginal cost, $MC_j$, is given by

$$MC_j = c_j - e \cdot (d_j + a_j - 1), \tag{20}$$

where $a_j$ is the choice of opening another store ($a_j = 1$) or not ($a_j = 0$), and $d_j$ is the number of stores at the beginning of the period. The inverse demand is given by

$$p_t = b_0 - \frac{b_1}{x_t} Q_t, \tag{21}$$

where $Q_t$ is aggregate output, $x_t$ is an exogenous market state, and $p_t$ is the product price. We assume $x_t$ to be fixed at $\bar{x}$ in the following. This is a slight simplification relative to Aguirregabiria et al. (2009), and I discuss the implications of relaxing this assumption in

subsection 3.3. Solving the Cournot model, I get the variable profits each period to be[5]

$$VP_j = \frac{\bar{x}}{b_1}\left(\frac{b_0 + MC_{-j} - 2MC_j}{3}\right)^2. \tag{22}$$

In addition to this variable cost, there is an overall fixed cost of being in the market, and a fixed cost that depends on the number of stores the firm has in the market. The former can be thought of as market specific logistics investments and marketing, and the latter can be thought of as fixed costs comming from specific stores. The functional form is

$$FC(d, a_j) = \theta^{FC_1} \cdot 1\{d_j + a_j - 1 > 0\} + \theta^{FC_2} \cdot (d_j + a_j - 1) + \theta^{FC_3} \cdot (d_j + a_j - 1)^2 \tag{23}$$

The total instantaneous profits given actions are then

$$\Pi_j(d, a_j, a_{-j}) = VP_j(d, a_j, a_{-j}) + FC(d, a_j). \tag{24}$$

The expected profits are simply the profits implied by different actions multiplied by their respective probabilities given the strategy of the other player plus an extreme value type 1 shock with scale parameter $\eta$.

# 3 The solutions

With a model class in hand and a specific model to motivate examples, we look at sub-stage solutions for some different types of games. They will differ by the number of actions, the state space structure, as well as the number of players. We start with the simplest binary duopoly game without exogenous states. Then we extend and generalize this case to that of a multivariate model, a case with non-directional exogenous states, and the case with more than two players. Further extensions and combinations of the different features are possible. However, in this paper the goal is to make the steps sufficiently clear, that it is straight forward to derive the equilibrium conditions for similar models. This could be models with different transition structure or models with sequentially played directional games combined into one. For example, the model in Iskhakov et al. (2018) also derived the equilibrium conditions for the binary choice duopoly game with alternating moves, which turns out to lead to even simpler conditions than the ones below, because the expected value functions are less complicated to evaluate. One one player can affect their own directional state per turn.[6]

[5] In the reference model there appears to be a missing factor of 2 on the firm's own marginal cost and the variable profits expression is not consistent with the demand function. The inverse demand function is used instead to give an expression that's consistent with the multiplicative term outside of the parentheses.

[6] They are actually even able to show that there can be a unique equilibrium of the entire game if the game-wide technology state in their model evolves deterministically.

Common for all sub-classes of models we will solve are, that they start from the equilibrium conditions that all choice probabilities are given by the choice probability equations. This might seem like a stupid idea, because I mentioned earlier that each agent's solution is fully characterized by their integrated value function. The system of Bellman equations for a given sub-stage then has one equation per agent if there are no exogenous states or $|\mathcal{X}(d)|$ equations per agent if there are. If we solve it in the choice probability space we now have $J_i(d) - 1$ times as many. The last probability is given residually. The problem with value functions is that the system can have many solutions and that the fixed point problem is solved in unconstrained variables. If we solve it in the choice probability space instead our variables of interest are on a compact set and in section 4 I explain how this can be exploited to solve for all equilibria. The complete information games are especially convenient, as they are systems of multivariate polynomials for which there are robust algebraic and numerical all-solution methods. The general approach to deriving the conditions are explained in Appendix A. In the corner and edge states only one player can choose and action, so we solve the problem in value function space.

## 3.1 Binary duopoly game

In the binary duopoly game, each player has the choice of opening a new store or not. If their states are absorbing their choice set is empty. To solve the game we need equilibrium conditions for the binary duopoly game. I first need to solve the corner states. Sometimes it's easier to always give all players the full choice set, even in the absorbing state as in Aguirregabiria et al. (2009). The value in such a state where both agents have a choice but they are in their absorbing state, $\bar{d} = \left(\bar{d}_1, \bar{d}_2\right)$, is

$$\begin{aligned}
\mathrm{IV}_i(\bar{d}) &= \eta \log\left(\exp\left(\frac{v_i^1(\bar{d})}{\eta}\right) + \exp\left(\frac{v_i^2(\bar{d})}{\eta}\right)\right) \Leftrightarrow \\
\mathrm{IV}_i(\bar{d}) &= \frac{\eta \log(2) + \Pi_i(\bar{d}, 1, 1)}{1 - \beta_i}
\end{aligned}$$

This is simply the usual infinitely discounted profits, but with the addition of $\eta \log(2)$. This is the expected value of the shock given optimal behavior. It is well-known from the CCP literature, see for example Hotz and Miller (1993); Aguirregabiria and Mira (2002). There, the expression is $-\log(p_1^1)$, which is in line with our expression, as they have a unit scale parameter, and the optimal choice probability is obviously 0.5. Since profit is the same no matter the choice, it is possible for the agent to simply choose the choice with the biggest shock in a given realization.

The choice probability of 0.5 follows directly from the conditional choice probability for-

mula and this feature is inherent to the class of models. With the taste shocks, we're not talking about mixed strategies, but rather so-called behavioral strategies. This is because all taste shocks are *known* to each agent. As a result, we could not have "mixing" strategies of 0.3 in the corner state(s) in this model. They will mechanically be 0.5 or else the agent will fail to take advantage of the taste shock arbitrage.

Alternatively, we give only agents a choice of doing nothing and get

$$\begin{aligned}
\text{IV}_i(\bar{s}) &= \eta \log\left(\exp\left(\frac{\Pi_i(\bar{d},1,1)+\beta_i \text{IV}_i(\bar{d})}{\eta}\right)\right) \\
\text{IV}_i(\bar{s}) &= \frac{\Pi_i(\bar{d})}{1-\beta_i}
\end{aligned}$$

which is the discounted stream of infinite profits an agent gets in the corner. We have not removed the taste shocks, but since it has mean zero, and there is no way to avoid bad shocks, they don't enter the discounted profits. We've solved the corner states.

In the *edge states*, one player has no choice but the other player can still decide to increment their state. The solution then changes, even for the playing who's constrained. Say player 1 is in the absorbing state then $d_1^e = (\bar{d}_1, d_2)$ and

$$\text{IV}_1(d_1^e) = \eta \log\left(\exp\left(\frac{p_2^1(d_1^e)(\Pi_1(d_1^e,1,1)+\beta_1 \text{IV}_1(d_1^e)) + p_2^2(d_1^e)\tau_1(d_1^e,1,2)}{\eta}\right)\right) \tag{25}$$

$$= p_2^1(d_1^e)(\Pi_1(d_1^e,1,1)+\beta_1 \text{IV}_1(d_1^e)) + p_2^2(d_1^e)\tau_1(d_1^e,1,2) \Leftrightarrow \tag{26}$$

$$\text{IV}_1(d_1^e) = \frac{p_2^1(d_1^e)\Pi_1(d_1^e,1,1) + p_2^2(d_1^e)\tau_1(d_1^e,1,2)}{1-p_2^1(d_1^e)\beta_1} \tag{27}$$

We see the value of waiting is much like before, but now there's a chance to move out of the current state. This can be seen in the numerator, but also in the denominator. It is not only $1-\beta_1$, but there is a correction for the chance that we will stay in $d_1^e$. If the probability is very high, then it's close to the infinite stream. If it is low, then we will probably just get the profits and continuation value from the other possible state, and the denominator is just 1. This correction is of course the choice probability that the other player chooses to stay in their current state.

The solution for player one depends on a choice probability of player 2: $p_2^1(d_1^e)$. From player two's perspective, this is just a normal single agent dynamic programming problem with an infinite horizon. The solution is found following the literature on single agent models by solving the following fixed point problem:

$$\Gamma(\text{IV}_2(d_1^e)) = \eta \log\left(\exp\left(\frac{\Pi_2(d_1^e,1,1)+\beta_2 \text{IV}_2(d_1^e)}{\eta}\right) + \exp\left(\frac{\tau_2(d_1^e,1,2)}{\eta}\right)\right) \tag{28}$$

and $\Gamma(v)$ is a contraction and we can safely use function iteration or Newton's method.[7] We can of course also use any other root-finding algorithm. From the fixed-point we can find the probabilities, and then we have the solution. If player two is constrained and player one is not, then we can simply flip indices, introduce $d_2^e = (d_1, \bar{d}_2)$, and get

$$\mathrm{IV}_2(d_2^e) = \frac{p_1^1(d_2^e)\Pi_2(d_2^e, 1, 1) + p_1^2(d_2^e)\tau_2(d_2^e, 2, 1)}{1 - p_1^1(d_2^e)\beta_2} \tag{29}$$

$$\mathrm{IV}_1(d_2^e) = \eta \log\left(\exp\left(\frac{\Pi_1(d_2^e, 1, 1) + \beta_1 \mathrm{IV}_1(d_2^e)}{\eta}\right) + \exp\left(\frac{\tau_1(d_2^e, 2, 1)}{\eta}\right)\right) \tag{30}$$

So there is only one equilibrium in the corner state and in the edge states. This is not surprising either, since the edges are single agent dynamic programming problems. In the edge cases we can use the complete information solution as the starting value for the value function search. As suggested earlier, the real interesting situation comes from interior states.

In the interior states, we get a coupled system of Bellman equations expressed by the Bellman operators

$$IV_1(d) = \eta \log\Big[\exp\left(\frac{p_2^1(d)\left(\Pi_1(d, 1, 1) + \beta_1 V_1(d)\right) + p_2^2(d)\tau_1(d, 1, 2)}{\eta}\right) + \tag{31}$$

$$\exp\left(\frac{p_2^1(d)\tau_1(d, 2, 1) + p_2^2(d)\tau_i(d, 2, 2)}{\eta}\right)\Big] \tag{32}$$

$$IV_2(d) = \eta \log\Big[\exp\left(\frac{p_1^1(s)\left(\Pi_2(d, 1, 1) + \beta_2 V_2(d)\right) + p_1^2(d)\tau_2(d, 2, 1)}{\eta}\right) + \tag{33}$$

$$\exp\left(\frac{p_1^1(d)\tau_2(d, 1, 2) + p_1^2(d)\tau_2(d, 2, 2)}{\eta}\right)\Big] \tag{34}$$

If we consider the right hand sides as functions of $IV_1$ and $IV_2$ respectively, they are contractions given $p_2^1(d)$ in the first line and the second line is a contraction given a $p_1^1(d)$. The system as a hwole is generally *not* a contraction. Then, fixed point iterations cannot generally find all equilibria. Luckily, I can take advantage of the content of Appendix A, and get expressions that characterize the solution of the system. In Appendix B I show that the coupled Bellman equations can be solved in the space of choice probabilities by solving the following system

$$0 = K_1^1 - \eta \log\left(\frac{p_1^1(d)}{1 - p_1^1(d)}\right) + \left(K_1^2 - \beta_1 \eta \log\left[1 - p_1^1(d)\right]\right) p_2^1(d) + K_1^3 p_2^2(d)^2 \tag{35}$$

$$0 = K_2^1 - \eta \log\left(\frac{p_2^1(d)}{1 - p_2^1(d)}\right) + \left(K_2^2 - \beta_2 \eta \log\left[1 - p_2^1(d)\right]\right) p_1^1(d) + K_2^3 p_1^2(d)^2 \tag{36}$$

[7] The contractive properties follows from the single agent nature of these states since player one has no actions to take.

with the following constants

$$
\begin{aligned}
K_1^1 &\equiv [\tau_1(d,1,2) - \tau_1(d,2,2)] \\
K_2^1 &\equiv [\tau_2(d,2,1) - \tau_2(d,2,2)] \\
K_1^2 &\equiv \Big[\Pi_1(d,1,1) + \tau_1(d,2,2) - \tau_1(d,2,1) - \tau_1(d,1,2)\Big] + \beta_1\tau_1(d,2,2) \\
K_2^2 &\equiv \Big[\Pi_2(d,1,1) + \tau_2(d,2,2) - \tau_2(d,1,2) - \tau_2(d,2,1)\Big] + \beta_2\tau_2(d,2,2) \\
K_1^3 &\equiv \beta_1\left(\tau_1(d,2,1) - \tau_1(d,2,2)\right) \\
K_2^3 &\equiv \beta_2\left(\tau_2(d,1,2) - \tau_2(d,2,2)\right)
\end{aligned}
$$

This gives us a system of two equations in two unknowns. These equations are sufficient to find the interior solutions for the strategies, and depend only on things we know once we reach the sub-stage that contains $d$.

### 3.1.1 Complete information

The complete information case obtains, as we let $\eta \to_+ 0$. In corner cases we get that

$$\mathrm{IV}_1(\bar{d}) = \frac{\Pi_1(\bar{d})}{1-\beta_1} \tag{37}$$

$$\mathrm{IV}_2(\bar{d}) = \frac{\Pi_2(\bar{d})}{1-\beta_2} \tag{38}$$

where $\Pi_i(\bar{d})$ is the profits pertaining to whatever choice we have chosen to be standard in the corner state. If it is really important to have two choices here, then it is simply the larger of the two entering the numerator, and if there's a tie, a tie-breaking rule must be applied. In the player 1 edge cases, we get

$$\mathrm{IV}_1(d) = \frac{p_1^1(d)\Pi_1(d,1,1) + p_2^1(d)\tau_1(d,1,2)}{1 - p_2^1(d)\beta_1} \tag{39}$$

$$\mathrm{IV}_2(d) = \max\left\{\frac{\Pi_2(d,1,1)}{1-\beta_2}, \tau_2(d,1,2)\right\} \tag{40}$$

For player 2's edge cases, we get

$$\mathrm{IV}_1(d) = \max\left\{\frac{\Pi_1(d,1,1)}{1-\beta_1}, \tau_1(d,2,1)\right\} \tag{41}$$

$$\mathrm{IV}_2(d) = \frac{p_1^1(d)\Pi_2(d,1,1) + p_1^2(d)\tau_2(d,2,1)}{1 - p_1^1(d)\beta_2}. \tag{42}$$

Now, we need to solve the case where there can potentially be several equilibria. To solve the interior points, we have to solve systems of polynomials. Luckily, they are very simple in

the binary case:

$$\begin{aligned}
0 =& \tau_1(d,1,2) - \tau_1(d,2,2) + \\
& \left[\Pi_1(d,1,1) - \tau_1(d,2,1) - \tau_1(d,1,2) + (1+\beta_1)\tau_1(d,2,2)\right] p_2^1(d) + \\
& \beta_1 \left[\tau_1(d,2,1) - \tau_1(d,2,2)\right] p_2^1(d)^2 \\
0 =& \tau_2(d,2,1) - \tau_2(d,2,2) + \\
& \left[\Pi_2(d,1,1) - \tau_2(d,1,2) - \tau_2(d,2,1) + (1+\beta_2)\tau_2(d,2,2)\right] p_1^1(d) + \\
& \beta_2 \left[\tau_2(d,1,2) - \tau_2(d,2,2)\right] p_1^1(d)^2
\end{aligned}$$

We can solve each equation by itself, and look for solutions in the unit square. This is simple application of the usual formula for the roots of a second-order polynomial. However, we could also use the methods in section 4.

That was the mixed strategies. In the complete information case, we also have pure strategies to consider. These might be a bit more tedious to find, but essentially there can only be four possible solutions: $(p_1^1, p_2^1) \in \{(0,1),(1,0),(0,0),(1,1)\}$. Say $p_2^1 = 0$ such that action 1 is certain. Then we get

$$v_1^1(d) = \Pi(d,1,2) + \beta \mathrm{IV}\left((d_1, d_2+1)\right) \tag{43}$$

$$v_1^2(d) = \Pi(d,2,2) + \beta \mathrm{IV}\left((d_1+1, d_1+1)\right) \tag{44}$$

Everything here is known, so we can simply calculate the best response of player 1 to player two playing $p_2^1 = 0$:

$$(p_1^1)^*(p_2^1 = 0) = \begin{cases} 1 & \text{if } v_1^1(d) \geq v_1^2(d) \\ 0 & \text{if } v_1^2(d) > v_1^1(s) \end{cases} \tag{45}$$

similarly for player 2

$$(p_2^1)^*(p_1^1 = 0) = \begin{cases} 1 & \text{if } v_2^1(s) \geq v_2^2(s) \\ 0 & \text{if } v_2^2(d) > v_2^1(s) \end{cases} \tag{46}$$

This tells us that we can easily verify if $(p_1^1, p_2^1) \in \{(0,0),(0,1),(1,0)\}$ is a solution. The pair $(p_1^1, p_2^1) = (1,1)$ is a solution if:

$$\frac{\Pi_1(d,1,1)}{1-\beta_1} > \tau_1(d,2,1) \tag{47}$$

$$\frac{\Pi_2(d,1,1)}{1-\beta_2} > \tau_2(d,1,2) \tag{48}$$

This gives us a quite easy way to look for pure strategy equilibria.

The procedure for the complete information game is now complete. It will cycle through all sub-stages, looking for pure strategy and mixed strategy equilibria in all sub-stages.

## 3.2 Multinomial duopoly game

Binary games are popular and I showed that the simple choice structure can simplify the solution a lot. In this section, I show that the same type of equilibrium conditions emerges multinomial case and that I can use the same strategy to find them.

In this section, $\mathcal{A}_i(d)$ can contain more than two elements. In all states where there are only two choices, we can simply use the conditions from the previous section. If there are three or more, we will need to derive new conditions. We only have $J_i(d)-1$ free probabilities for each player due to the unit sum restriction, and we write $p_i^J(d) \equiv 1 - \Sigma p_i^{-J_i(d)}$, where $\Sigma p_i^{-J_i(d)} = \sum_{j=1}^{J_i(d)-1} p_i^j$.

The structure of the equilibrium conditions is familiar, but this time the solution is characterized by system of $N_E = \sum_{i=1}^{\mathcal{N}}(J_i(d)-1)$ equations in $N_E$ unknowns: the sum of $J_i(d)-1$ free probabilities across all players. There are potentially also many more terms. Define $\widehat{J}_i \equiv J_i(d)$ for compactness. In Appendix C I derive the equilibrium conditions and the constant definitions can also be found there. There is one condition for each player with the following form

$$\begin{aligned}
0 = {} & L_1^1 - \eta \log\left(\frac{p_1^1(d)}{p_1^{\widehat{J}_1}(d)}\right) \\
& + \left(L_1^2 - \beta_1 \eta \log\left(p_1^{\widehat{J}_1}\right)\right) p_2^1(d) + \beta_1 p_2^1(d) \sum_{k=1}^{\widehat{J}_2-1} L_1^{3,k} p_2^k(d) + \sum_{k=2}^{\widehat{J}_2-1} L_1^{4,k} p_2^k(d) \qquad (49) \\
0 = {} & L_2^1 - \eta \log\left(\frac{p_2^1(d)}{p_2^{\widehat{J}_2}(d)}\right) \\
& + \left(L_2^2 - \beta_2 \eta \log\left(p_2^{\widehat{J}_2}\right)\right) p_1^1(d) + \beta_2 p_1^1(d) \sum_{k=1}^{\widehat{J}_1-1} L_2^{3,k} p_1^k(d) + \sum_{k=2}^{\widehat{J}_1-1} L_2^{4,k} p_1^k(d) \qquad (50)
\end{aligned}$$

Again, the goal here was to express the behavior strategies as functions of things that are known at this sub-stage. The remaining equilibrium equations indexed $j_1 \in \{2, \ldots J_1(d)-1\}$ and $j_2 \in \{2, \ldots, J_2(d)-1\}$ are

$$0 = L_1^5 - \eta \log\left(\frac{p_1^{j_1}(d)}{p_1^{\widehat{J}_1}(d)}\right) + \sum_{k=1}^{\widehat{J}_2-1} L_1^{6,k} p_2^k(d) \qquad (51)$$

$$0 = L_2^5 - \eta \log\left(\frac{p_2^{j_2}(d)}{p_2^{\widehat{J}_2}(d)}\right) + \sum_{k=1}^{\widehat{J}_2-1} L_2^{6,k} p_1^k(d) \qquad (52)$$

$$(53)$$

These are much simpler and the products between strategy elements are of lower order. This is because they are derived from equilibrium conditions that characterize actions that force

at least one element in $d$ to transition. Of course, $(p_1^1(d), p_2^1(d))$ still enter, so we cannot solve them independently of the first more complicated set of equations.

In many practical examples, we expect $J_i(d)$ to be relatively small. Increased computational cost of course comes with the explicit advantage that we can consider much more complicated interactions between agents. More choices add the complexity and flexibility copmared to the binomial models. Being able to add more nuance than "build/don't build", "enter/don't enter", "invest/don't invest" can the usefulness of these models in an empirical context.

### 3.2.1 Complete information

This case is more complicated than before. Specifically, none of the equations are simple second-order polynomials in a single choice probability for any player. Rather, we have a second-order multivariate polynomials. However, it is still true that the deterministic model can be solved completely with the methods we will see later. All that is needed is that we have a system of polynomials, and the limiting system is exactly that. Besides the mixed strategy equilibria, we have to find the pure strategy equilibria. The procedure is the same as in the duopoly case. This time there are many more cases, so there is a curse of dimensionality in this part of the solution approach, but this part in negligible as it only requires simple comparisons.

## 3.3 Binary duopoly game with non-directional exogenous states

An interesting extension is to include non-directional exogenous states. Things like market activity states (GDP growth, unemployment, inflation, and so on) is very relevant for industry dynamics, and failing to account for them might give inaccurate conclusions about market dynamics. As for the other extensions presented here, the approach is more or less the same, but the multiple states per sub-stage do come with a complication. In Iskhakov et al. (2018) they actually have a non-directional exogenous state in their alternating move game: the indicator for whose turn it is. The expressions end up very simple because on player has the empty set as their action space each turn. Here, we do not impose the same restriction which adds some complications in terms of having a chance of staying in the same directional state from one turn to the next but having the non-directional exogenous state changing.

In fig. 3 we see a simple state space with an exogenous non-directional state $x$ that takes values in $\mathcal{X} = \{x_1, x_2\}$, and a directional state $s$ that takes values in $\mathcal{D} = \{d_1, d_2, d_3\}$. Here, we need to take into consideration, that we can stay in a directional state for several periods, but the non-directional state can change. Black arrows are cases where there are only non-

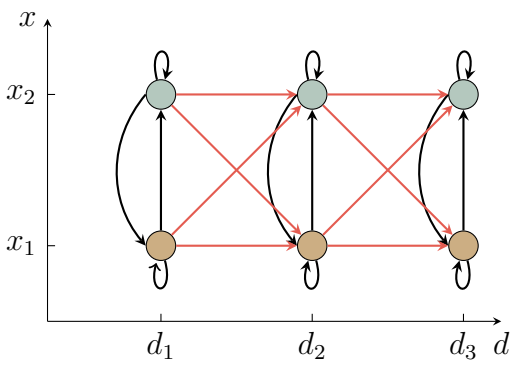



Figure 3: A simple state space with a directional state, $d \in \mathcal{D}$, and non-directional state, $x \in \mathcal{X}$.

directional moves, and red arrows show moves of directional character. We see that both the directional transitions today as well as the non-directional transitions have to be considered in each sub-stage.

This time around we have $2 \times |\mathcal{X}|$ free strategy probabilities. I derive them in Appendix E where the constant definitions can also be found. For $x \in \mathcal{X}$ they are

$$0 = Q_1^1 - \frac{\log\left[p_1^1(d,x)\right]}{1-\log\left[p_1^1(d,x)\right]} + \left[Q_1^2 - \beta_1\eta \sum_{x' \in \mathcal{X}} \pi^{\mathcal{X}}(x'|x) \log\left[p_1^2(d,x')\right]\right] p_2^1(d,x) + p_2^1(d,x) \sum_{x' \in \mathcal{X}} p_2^1(d,x') Q_1^{3,x'} \tag{54}$$

$$0 = Q_2^1 - \frac{\log\left[p_2^1(d,x)\right]}{1-\log\left[p_2^1(d,x)\right]} + \left[Q_2^2 - \beta_2\eta \sum_{x' \in \mathcal{X}} \pi^{\mathcal{X}}(x'|x) \log\left[p_2^2(d,x')\right]\right] p_1^1(d,x) + p_1^1(d,x) \sum_{x' \in \mathcal{X}} p_1^1(d,x') Q_2^{3,x'} \tag{55}$$

$$\tag{56}$$

Once again we're in a multivariate polynomial case as $\eta$ goes to zero.

### 3.3.1 Complete information case

Given the description above, it is quite easy to solve the corner states, although we need to find a value function per $x \in \mathcal{X}$. The value functions are given by the following expression

$$\forall x \in \mathcal{X} : \mathrm{IV}_i(d,x) = \Pi_i(d,x,1,1) + \beta_i \sum_{x' \in |X|} \pi(x'|d,x) \mathrm{IV}_i(d,x') \tag{57}$$

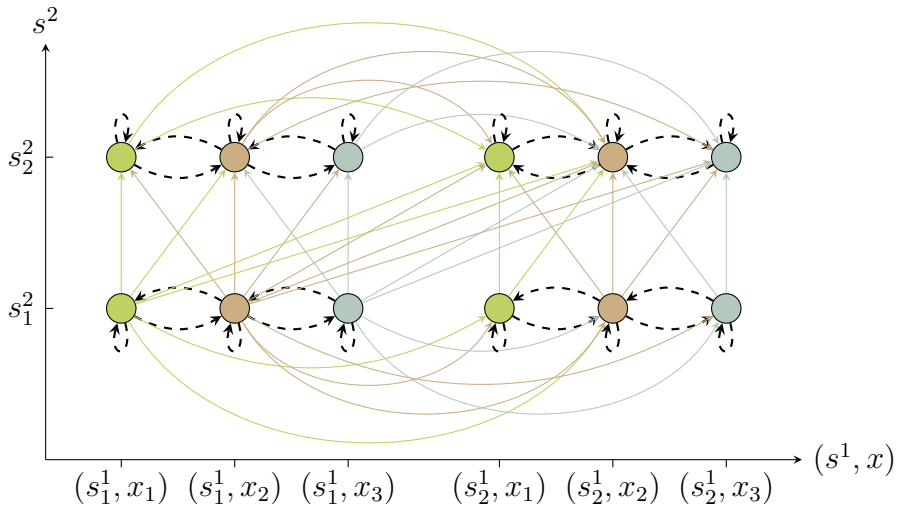



Figure 4: A partial view of the state space of the expansion game with non-directional exogenous states.

Let $\text{IV}_i$ be the $\text{IV}_i(x)$'s stacked in a vector, and similarly for $\Pi$, and let $F$ be the transition matrix for the non-directional state, then we can write the above as

$$\text{IV}_i = \Pi + \beta_i F \text{IV}_i \tag{58}$$

so we can write

$$\text{IV}_i = (I - \beta_i F)^{-1} \Pi \tag{59}$$

The edge states are solved much like in the binary duopoly. Of course, we also have a vector of value functions to find here. For player 1 at the edge where $d_1 = \bar{d}_1$ we have the following at a given $x \in \mathcal{X}$

$$\forall x \in \mathcal{X} : \text{IV}_1(d_1^e, x) = p_2^1(d_1^e, x)\hat{\tau}_1(d_1^e, x, 1, 1) + p_2^2(d_1^e, x))\hat{\tau}_1(d_1^e, x, 1, 2) \tag{60}$$

if we use $*$ to denote the Hadamard product, we can re-write the above to vector-matrix notation as before we have for $x \in \mathcal{X}$

$$\begin{aligned}
&\text{IV}_1(d_1^e, x) = p_2^1(d_1^e, x)\Pi_1(d_1^e, x, 1, 1) + p_2^1(d_1^e, x)\beta_1 \sum_{x' \in \mathcal{X}} \pi(x'|d, x)\text{IV}_1(d_1^e, x') + p_2^1(x)\hat{\tau}(d, x, 1, 2) \Leftrightarrow \\
&\text{IV}_1(d_1^e) = P_2^1(d_1^e) * \Pi_1(d, 1, 1) + (\vec{1} - P_2^1(d_1^e)) * \hat{\tau}(d_1^e, 1, 2) + \beta_1 P_2^1(d_1^e) * F\text{IV}_1(d_1^e) \Leftrightarrow \\
&\text{IV}_1(d_1^e) = (I - \beta_1 P_2^1(d_1^e) * F)^{-1}(P_2^1 * \Pi_1(d_1^e, 1, 1) + (\vec{1} - P_2^1(d_1^e)) * \hat{\tau}(d_1^e, 1, 2))
\end{aligned} \tag{61}$$

where $\vec{1}$ is an appropriately dimensioned vector of ones, and $P_i^j(d)$ is a vector for subject $i$ of choice probabilities for choice $j$ across all $x \in \mathcal{X}$. As before, we need to find $P_2^1(d)$ to

calculate this, but everything else is known. For player 2 at this edge, we get the Bellman equation

$$\mathrm{IV}_2(d_1^e) = \max\left(\Pi(d_1^e, 1, 1) + \beta_2 F \mathrm{IV}_2(d_1^e), \Pi(d_1^e, 1, 2) + \beta_2 F \mathrm{IV}_2((\bar{d}_1, d_2 + 1))\right) \tag{62}$$

again in vector-matrix notation. We have to resort to some fixed-point scheme, and value function iterations works because this is a single agent model as in the other cases.

With the corner and edges solved, we are ready to consider the interior sub-stages. These are more complicated, as we have $4^{|\mathcal{X}|}$ different cases to check for pure strategies, four per exogenous state, as well as the mixed strategy equilibria from the system of polynomials. This can potentially be a lot of cases, but luckily each case is relatively cheap to check. All that has to be done is to loop through all the possible pure strategies, calculate the value conditional on those strategies, and compare to find the best responses. The above description and associated examples hopefully show that solving directional games exogenous non-directional states is certainly possible, and practically feasible. Some of the details are significantly different from the simpler cases, but there is nothing so complicated it cannot be handled using some algebra, and knowledge of the model.

## 3.4 Binary triopoly game

The last of the extensions we will see is the oligopoly version that goes beyond simply duopoly. For three players with binary choice, we get three equations in three unknowns. The derivations for the interior solutions are in Appendix D where the constant definitions can also be found. The basic structure is again the same, and the resulting equations are:

$$0 = M_1^1 - \eta \log\left[\frac{p_1^1(d)}{1 - p_1^1(d)}\right] + p_2^1(d) M_1^2 + p_3^1(d) M_1^3 + \left[M_1^4 - \beta_1 \eta \log\left[p_1^2(d)\right]\right] p_2^1(d) p_3^1(d) \tag{63}$$

$$+ M_1^5 p_2^1(d)^2 p_3^1(d) + M_1^6 p_2^1(d) p_3^1(d)^2 + M_1^7 p_2^1(d)^2 p_3^1(d)^2 \tag{64}$$

$$0 = M_2^1 - \eta \log\left[\frac{p_2^1(d)}{1 - p_2^1(d)}\right] + p_1^1(d) M_2^2 + p_3^1(d) M_2^3 + \left[M_2^4 - \beta_1 \eta \log\left[p_2^2(d)\right]\right] p_1^1(d) p_3^1(d) \tag{65}$$

$$+ M_2^5 p_1^1(d)^2 p_3^1(d) + M_2^6 p_1^1(d) p_3^1(d)^2 + M_2^7 p_1^1(d)^2 p_3^1(d)^2 \tag{66}$$

$$0 = M_3^1 - \eta \log\left[\frac{p_3^1(d)}{1 - p_3^2(d)}\right] + p_2^1(d) M_1^2 + p_1^1(d) M_3^3 + \left[M_3^4 - \beta_3 \eta \log\left[p_3^2(d)\right]\right] p_1^1(d) p_2^1(d) \tag{67}$$

$$+ M_3^5 p_2^1(d)^2 p_1^1(d) + M_3^6 p_2^1(d) p_1^1(d)^2 + M_3^7 p_2^1(d)^2 p_1^1(d)^2 \tag{68}$$

These are only the interior equilibrium conditions. What about the corners and edge states? We could repeat everything, but luckily we've done most of the work already. If all players are at terminal directional nodes, then we're in a corner state. If we are in an edge state

where one player is at a terminal directional node, then the remaining two players simply play a "two-player" interior game. If two players are at terminal nodes, the remaining player is simply in a situation like the "two-player" edge game. Else, we're at an interior sub-stage, and we've just seen how to find equilibria there.

It is quite interesting that a three-player game in this class has such simple equilibrium conditions. Many empirical models of interactions between firms can be greatly enhanced if there can be more than two players in the game. The results here are for three players, but there is no reason why we couldn't solve the general case as in the multinomial choices duopoly. The expressions would only become messier.

# 4 Solving the equilibrium conditions

In this section, we will introduce the specific methods that we will use to find the non-pure strategies. For the complete information games, we will use homotopy methods for systems of polynomial equations. In this case, we exploit structure to get a stable algorithm. For the incomplete information games, we will use interval arithmetic. In this case, there is no exploitable structure since the systems are only *almost* polynomials - logarithmic terms appear in the coefficients.

## 4.1 Systems of multivariate polynomial equations

Homotopy continuation methods are based around sequentially solving $H(x,t)$ for $x$ given a $t$, where $H$ is called a homotopy. In our case it allows us to solve our system of multivariate polynomials equations $F$ by first solving another system $G$ that is related to $F$ in some way. We choose $G$ such that it is easy to solve and exploit that our homotopy will have the property that

$$H(x,1) = G(x)$$
$$H(x,0) = F(x).$$

One choice of homotopy is then obviously

$$H(x,t) = (1-t)tF(x) + (t)G(x),$$

but others exist. The system $G$ is called a start system, as is supposed to be easy to solve such that the homotopy continuation method can follow the path of solutions starting from those of $G$ to those of $F$. Since our systems are not too large and not too sparse we use the

total degree start system. This means that $G$ has the form of:

$$G(x) = (x_1^{d_1} - a_1, x_2^{d_2} - a_2, \ldots, x_n^{d_n} - a_n)$$

where $d_i$ is the degree of the $i$th polynomial of $F$ and $a_i$ is a random number. The solutions to this starting system is trivial but allows us to slowly move towards the solutions of $F$. By adjusting $t$ in small increments, we can ensure that we have good starting values for each $t$. However, we would always wish to take as large steps as possible to save time spent. Predictor-corrector steps allow us to adjust the solver efficiently along the continuation path.

Homotopy continuation methods are numerical in nature, but for some classes of systems, they can be very efficient and accurate. Systems of polynomial equations are such an example. Nonetheless, any application needs to use a high-quality implementation, so I will use the HomotopyContinuation.jl package described in Breiding and Timme (2018) for the Julia Programming Language (Bezanson et al. (2017)).[8]

Gröbner bases methods: advantages are symbolic and parametric. Potentially you could trace out the solutions to a parametric family of solutions.

## 4.2 Interval root-finding methods

The homotopy methods may be less reliable for systems that are not polynomials and systems that do not have the structure we can exploit. An interesting alternative to consider is then interval arithmetic methods. An introduction to the methods, in general, can be found in Moore et al. (2009) and a brief introduction to the content and history of the IEEE 1788-2015 Standard for Interval Arithmetic can be found in Revol (2017) and Kubica (2019). The latter also covers relevant software and libraries. The lack of a standard before 2015 has probably meant that these methods are less widespread, as several edge-cases were implemented differently in different libraries which means that the user has to be more careful when interpreting the results.

Software such as INTLAB (Rump (1999)) for Matlab has been available for many years. Nonetheless, adoption in economics and also other disciplines. In economics some applications of these techniques exist, but most are using only interval arithmetics for validated numerics, see Barker and Rocco S (2011), Choobineh and Behrens (1993), and Jerrell (1997b). For economists using interval methods for optimization see Jerrell (1997a). Most of these papers are from the 1990s which saw a renewed interest in the methods, but software was less readily

[8]Other libraries exist, but HomotopyContinuation.jl has the clear advantage that it was written in a performant high-level language which means it's flexible and also easy to use without sacrificing speed. An earlier version of this paper used an interface to PHCPack Verschelde (1999).

available, and as mentioned this was before the IEEE standard was defined. For a modern implementation of the interval arithmetic and root-finding methods we, use the JuliaIntervals suite of software to work with intervals.

Since this paper is not about implementing new interval methods we will not give a complete treatment of the subject here, but to fix ideas we will introduce the basic concepts of interval numbers, arithmetic and function evaluation. Interval arithmetic works with the real, closed intervals $X = [\underline{\mathrm{x}}, \overline{\mathrm{x}}]$ where $\underline{\mathrm{x}}$ and $\overline{\mathrm{x}}$ are real *numbers*. A vector of intervals, $V = (X_1, X_2, \ldots, X_n)$ is called a *box* because it generalizes the geometric concept of an interval (1D) to a rectangle (2D) and beyond. Interval *arithmetic* is then defined by the usual operations such as addition, subtraction, multiplication and division:

$$X + Y = [\underline{\mathrm{x}} + \underline{\mathrm{y}}, \overline{\mathrm{y}} + \overline{\mathrm{x}}] \tag{69}$$

$$X - Y = [\underline{\mathrm{x}} - \overline{\mathrm{y}}, \overline{\mathrm{x}} - \underline{\mathrm{y}}] \tag{70}$$

$$X \cdot Y = [\min(\underline{\mathrm{x}} \cdot \underline{\mathrm{y}}, \underline{\mathrm{x}} \cdot \overline{\mathrm{y}}, \overline{\mathrm{x}} \cdot \underline{\mathrm{y}}, \overline{\mathrm{x}} \cdot \underline{\mathrm{y}}), \max(\underline{\mathrm{x}} \cdot \underline{\mathrm{y}}, \underline{\mathrm{x}} \cdot \overline{\mathrm{y}}, \overline{\mathrm{x}} \cdot \underline{\mathrm{y}}, \overline{\mathrm{x}} \cdot \underline{\mathrm{y}})] \tag{71}$$

$$X/Y = [\underline{\mathrm{x}}, \overline{\mathrm{x}}] \cdot [1/\overline{\mathrm{y}}, 1/\underline{\mathrm{y}}] \text{ if } \underline{\mathrm{y}}, \overline{\mathrm{y}} \neq 0 \tag{72}$$

with proper edge-case handling when the divisor contains 0. See the IEEE standard for all the rules that have been agreed upon in terms of arithmetic edge-cases. Since we're going to be solving systems of non-linear equations it is interesting what it means to apply a function to an interval. For monotonically increasing functions with real domains we can define corresponding interval functions quite easily. If $f : \mathbb{R} \to \mathbb{R}$, then the corresponding interval function $f_i$ is defined as

$$f([\underline{\mathrm{x}}, \overline{\mathrm{x}}]) = [f(\underline{\mathrm{x}}), f(\overline{x})] \tag{73}$$

and monotonically decreasing functions are handled using the additive inverse implied by the rules above. When passing intervals through functions composed of several functions we simply apply the functions in turn. Periodic functions such as many trigonometric functions are handled by explicitly using information about the location of critical points, and general functions can be handled by Taylor expansions. However, we will only use polynomials and logarithmic functions in this paper, and the functions involved are standard in the litterature and software considered. Below we will need a function that calculates the central element in an interval:

$$mid(X) = \frac{(\underline{\mathrm{x}} + \overline{\mathrm{x}})}{2}$$

with details regarding the handling of extended reals found in (IEEE, 2015, p. 64)

To solve interval root finding problems we need methods beyond rules for doing arithmetic and evaluations. To start looking for solutions the user has to provide an initial box. There

are two possibilities here. Either the box contains no solution or it contains at least one solution. If it contains no solution we would like to know, and if it contains solutions we would like to find them all. For multivariate root-finding, we can use the Krawczyk method which has Newton's method for intervals as a special case. One step in the iterative procedure starting from an initial box $X_0$ and where $X_k$ is the box we want to apply the iterative procedure to is defined as:

$$K(X) = y - Y \cdot f(y) + (I - Y \cdot F'(X)) \cdot (X - y) \tag{74}$$

$$X_{k+1} = K(X_k) \cap X_k \tag{75}$$

where $Y$ is some non-singular real matrix that approximates the inverse of the Jacobian evaluated at $mid(X)$, $y$ is some real vector inside the box $X$. Newton's method obtains from this method by setting $Y$ equal to the actual inverse Jacobian. The vector $y$ is can be set to $mid(X)$ but doesn't have to. If the resulting $X_{k+1}$ contains multiple disjoint boxes we must then apply $K(\cdot)$ to each of them. Superficially, one might think that an alternative is then just to multi-start a lot of Newton's method solves. A major difference is that the interval method is quite systematically removing large regions of the domain where roots can lie, and as soon as it is clear that some region cannot have a flip of the sign, that box is removed from the next iterate $K(X_k) \cap X_k$. In a multi-start regime, we would have to start a lot of local solves and observe them either failing or converging to the same roots. However, since the basin of attraction can be arbitrarily small or have irregular contours it's not a very robust method and certainly expensive. Notice, that the Jacobian is evaluated at a regular real vector, not a box. As always, it can be calculated by hand, but we use the forward mode automatic differentiation as implemented in Revels et al. (2016).

Once we obtain a collection of intervals as the solution we can check if they all contain a unique solution. It turns out that there is a relatively easy check that can be made. Rump (1980) showed that if $N(X) \subset X$ for Newton's method and $K(x) \subset X$ for Krawczyk's method then there is a unique solution in $X$. The surprising implication is that upon successful convergence, we get a collection of intervals that are mathematically proven to be intervals that bound only unique solutions. The methods also do not exclude any roots from iteration to iteration, so we will also know that it is all of them. Now, this is of course constrained by the fact that we must only use functions that we can evaluate at intervals, and we still need sufficient precision to make all the calculations involved. However, if there are numerical issues with something like the log function near 0 it is possible to switch to multiple-precission arithmetic in Julia by using the provided BigFloat number type that wraps MPFR (Fousse et al. (2007)). Increasing the precision of the arithmetic of course increases the runtime of the root-finding procedure.

# 5 Examples

In this section, I present some illustrations of actual solutions to some of the variants of the game. In dynamic gamees there can often be many Markov Perfect Equilibria. The goal here is to find them all. We need to find them all because one randomly chosen equilibrium might not represent the set of all equilibria in terms of profit distribution and so on. If the researcher would want to use the maximum likelihood method to estimate parameters they need the solutions to construct the likelihood of observing the data. However, the many equilibria does not have to be very different in their market structure or profit distribution as can be seen from the discussion in Iskhakov et al. (2018). Throughout, I use the same set of parameters such that the models are quite comparable. The outcomes might of course be quite different between multinomial duopoly and binary triopoly.[9] I provide more details for the binary duopoly than the rest. In the interst of space not all versions will have the same level of details, but to provide a reference point to understand the examples with fewer details the first example will be relatively detailed.

The solution methods described in the previous section can both be used to solve the complete information games. I have only seen them finding the same equilibria which is encouraging in terms of robustness of the suggested methods. I do not do a thorough benchmark here, but do provide a few comparisons. For the incomplete information game, the log's in the equilibrium conditions provide some issues for the more complicated models. It is apparently hard to exclude many regions that clearly do not represent equilibria as seen from then large residual values when evaluating the system at the midpoints. The inability to exclude these irrelevant regions made it prohibitive to use this method beyond the binary duopoly case that did not appear to have these issues.

## 5.1 Binary duopoly

## 5.2 Complete information

Let us consider a case where there are two players. In the complete information version of the game solution in the corner state $\bar{d} = (3, 3)$ is

$$\text{IV}_1(\bar{d}) = \frac{\Pi_1(\bar{d})}{1 - \beta_1} = 340.56 \tag{76}$$

$$\text{IV}_2(\bar{d}) = \frac{\Pi_2(\bar{d})}{1 - \beta_2} = 340.56 \tag{77}$$

[9]The following model parameters are used $j \in \{1, 2\}$: $\bar{d}_j = 3$, $\beta_j = 0.95$ $\bar{x} = 1.60$, $b_0 = 5.0$, $b_1 = 0.3$, $c_j = 0.6$, $e = 0.1$, $FC_1 = 1.5$, $FC_2 = 0.1$, $FC_3 = 0.2$.

Since there are no choices, there can only be one equilibrium. Then, I can either solve all the states where $\sum_i d_i = 5$, or solve the edges. I solve the edges because they require the same steps to be solved. First, solve along $d = (d_1, 3)$ and then along $d = (3, d_2)$. There can only be one equilbrium in each state and the solution is for both players to keep opening restaurants along the edges. This increases their own profits, and decreases the profits of the other player. Looking at the integrated values, we get

$$\begin{aligned}
&\text{IV}_1(3,3) = 340.56 && \text{IV}_2(3,3) = 340.56 && \\
&\text{IV}_1(2,3) = 340.56 && \text{IV}_2(2,3) = 340.56 && p_1^1(2,3) = 0 \\
&\text{IV}_1(1,3) = 340.29 && \text{IV}_2(1,3) = 341.27 && p_1^1(1,3) = 0 \\
&\text{IV}_1(0,3) = 339.66 && \text{IV}_2(0,3) = 342.67 && p_1^1(0,3) = 0 \\
&\text{IV}_1(3,2) = 340.56 && \text{IV}_2(3,2) = 340.56 && p_2^1(3,2) = 0 \\
&\text{IV}_1(3,1) = 341.27 && \text{IV}_2(3,1) = 340.29 && p_2^1(3,1) = 0 \\
&\text{IV}_1(3,0) = 342.67 && \text{IV}_2(3,0) = 339.66 && p_2^1(3,0) = 0
\end{aligned}$$

The last state on each edge has the same continuation value as the corner because it is possible to instantly transision to the absorbing situation since there is no lag in construction.

The next step is to solve the interior states, starting with $d = (2,2)$ which is possible because we have solutions for $\{(3,3),(2,3),(3,2)\}$. Here, we have two pure strategy equilibria: either both expand or neither expand. If one player allows the other player to expand first, they essentially get a period with very low profits only for them to have the optimal choice of expanding in the next period. So either both expand right away, or they don't. However, if we look at the solutions coming out of the homotopy continuation solution, we see that there is also a mixed strategy. Appendix G shows how to specifically solve this case and the output it generates. The system has four real solutions

$$\begin{aligned}
&^*(p_1^1(2,2), p_2^1(2,2)) = (0.658, 0.658) && (p_1^1(2,2), p_2^1(2,2) = (0.658, -0.623) \\
&(p_1^1(2,2), p_2^1(2,2)) = (-0.623, -0.658) && (p_1^1(2,2), p_2^1(2,2) = (-0.623, 0.658)
\end{aligned}$$

Obviously, only one of these are feasible. This gives us three equilibria in this sub-stage, and from here on I the algorithm in Iskhakov et al. (2015) to exhaustively search for equilbria. It turn out, that if the players either both expand or play a mixed expansion strategy, then there are also three equilibria in the $d = (1,1)$ state, but if the two players agree on not expanding in $d = (2,2)$, then the only equilbrium in the $d = (1,1)$ sub-stage is to expand into $d = (2,2)$ and stay there for a higher continuation value than they would have gotten if they ended up in $d = (3,3)$. In total, there are 7 equilibria in this game. Below I show the

first equilibrium found

$\mathrm{IV}_1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 348.0 | 348.0 | 344.0 | 339.4 |
| 1 | 348.0 | 347.9 | 344.4 | 340.3 |
| 2 | 345.3 | 344.4 | 341.2 | 340.6 |
| 3 | 342.7 | 341.3 | 340.6 | 340.6 |

$p_1^1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1 | 1.00 | 0.86 | 0.00 | 0.00 |
| 2 | 1.00 | 1.00 | 0.66 | 0.00 |
| 3 | – | – | – | – |

$\mathrm{IV}_2$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 348.0 | 348.0 | 345.3 | 342.7 |
| 1 | 348.0 | 347.9 | 344.4 | 341.3 |
| 2 | 344.0 | 344.4 | 334.2 | 340.6 |
| 3 | 339.4 | 340.3 | 340.6 | 340.6 |

$p_2^1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0.00 | 1.00 | 1.00 | – |
| 1 | 0.00 | 0.86 | 1.00 | – |
| 2 | 0.00 | 0.00 | 0.66 | – |
| 3 | 0.00 | 0.00 | 0.00 | – |

$n_{EQ}$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 3 | 1 | 1 |
| 2 | 1 | 1 | 3 | 1 |
| 3 | 1 | 1 | 1 | 1 |

Where $n_{EQ}$ denotes the number of found equilibria at each sub-stage. For the case where the don't expand action is chosen by each player in $d = (2, 2)$ we get

$\mathrm{IV}_1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 348.7 | 348.0 | 348.0 | 339.4 |
| 1 | 349.4 | 348.7 | 348.7 | 340.3 |
| 2 | 349.4 | 348.7 | 348.7 | 340.6 |
| 3 | 342.7 | 341.3 | 340.6 | 340.6 |

$p_1^1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 1.0 | 1.0 | 1.0 | 0.0 |
| 3 | – | – | – | – |

$\mathrm{IV}_2$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 348.7 | 349.4 | 349.4 | 342.7 |
| 1 | 348.0 | 348.7 | 348.7 | 341.3 |
| 2 | 348.0 | 348.7 | 348.7 | 340.6 |
| 3 | 339.4 | 340.3 | 340.6 | 340.6 |

$p_2^1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0.0 | 0.0 | 1.0 | – |
| 1 | 0.0 | 0.0 | 1.0 | – |
| 2 | 0.0 | 0.0 | 1.0 | – |
| 3 | 0.0 | 0.0 | 0.0 | – |

$n_EQ$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 3 | 1 |
| 3 | 1 | 1 | 1 | 1 |

It is clear that in this equilibrium, the players specifically target the appropriate actions to eventually reach $(2, 2)$ and stay there.

## 5.3 Incomplete information

We solve the binary duopoly game using interval methods as described above. We set $\eta = 0.64$ to set it a some value, and loop through the states in the same way as above. The corner and edge states are described in section 3. As it turns out, there are only two solutions in this incomplete information game, or smoothed game. The first one can still be fully listed in this binary duopoly game and is

$\mathrm{IV}_1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 361.6 | 361.6 | 356.6 | 340.1 |
| 1 | 361.6 | 359.4 | 354.9 | 341.0 |
| 2 | 357.7 | 354.6 | 350.5 | 341.2 |
| 3 | 344.0 | 342.7 | 341.6 | 340.6 |

$p_1^1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | $4 \cdot 10^{-11}$ | $3 \cdot 10^{-13}$ | $1 \cdot 10^{-12}$ | $3 \cdot 10^{-12}$ |
| 1 | 0.9977 | 0.6749 | 0.3156 | 0.2351 |
| 2 | 0.9999 | 0.9994 | 0.6273 | 0.6201 |
| 3 | – | – | – | – |

$\mathrm{IV}_2$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 361.6 | 361.6 | 357.7 | 344.0 |
| 1 | 361.6 | 359.4 | 354.6 | 342.7 |
| 2 | 356.6 | 354.9 | 350.5 | 341.6 |
| 3 | 340.1 | 341.0 | 341.2 | 340.6 |

$p_2^1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | $4 \cdot 10^{-11}$ | 0.9977 | 0.9999 | – |
| 1 | $3 \cdot 10^{-13}$ | 0.6749 | 0.9994 | – |
| 2 | $1 \cdot 10^{-12}$ | 0.3156 | 0.6273 | – |
| 3 | $3 \cdot 10^{-11}$ | 0.2351 | 0.6201 | – |

$n_E Q$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 2 | 1 |
| 3 | 1 | 1 | 1 | 1 |

the second one is

$\mathrm{IV}_1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 360.0 | 360.0 | 348.0 | 339.4 |
| 1 | 360.0 | 357.7 | 348.7 | 340.3 |
| 2 | 355.3 | 352.0 | 348.7 | 340.6 |
| 3 | 344.0 | 342.7 | 340.6 | 340.6 |

$p_1^1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | $4 \cdot 10^{-11}$ | $8 \cdot 10^{-13}$ | $1 \cdot 10^{-12}$ | $3 \cdot 10^{-12}$ |
| 1 | 0.9993 | 0.7265 | 0.4085 | 0.2351 |
| 2 | 0.9999 | 0.9994 | 0.6273 | 0.6201 |
| 3 | – | – | – | – |

$\mathrm{IV}_2$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 348.7 | 349.4 | 349.4 | 342.7 |
| 1 | 348.0 | 348.7 | 348.7 | 341.3 |
| 2 | 348.0 | 348.7 | 348.7 | 340.6 |
| 3 | 339.4 | 340.3 | 340.6 | 340.6 |

$p_2^1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | $4 \cdot 10^{-11}$ | 0.9993 | 0.9999 | – |
| 1 | $8 \cdot 10^{-13}$ | 0.7265 | 0.9994 | – |
| 2 | $1 \cdot 10^{-12}$ | 0.4085 | 0.6273 | – |
| 3 | $3 \cdot 10^{-12}$ | 0.2351 | 0.6201 | – |

$n_E Q$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 2 | 1 |
| 3 | 1 | 1 | 1 | 1 |

Some of the choice probabilities are very close to pure strategies. If no stores are open, it's obviously a good idea to open, because the alternative is zero profits forever. For an agent to choose not to open, they'd have to draw a very significant taste shock to the action of not expanding, and there are choice probabilities on the order og $10^{-13}$ as a result. As mentioned in the introduction of the model, they reveal that our model is lacking a separate entry phase where this two player market begins to exist. Conditional on existing, it is simply profitable to open a restaurant. In the real world, there would of course be alternative investments to consider such as opening a bar, cafe or investing the money in stocks. We've implicitly assumed that the profits are net these alternative costs. It is possible to solve the model from $d = (4, 4)$ back to $d = (1, 1)$ if these dynamics are irrelevant since our model is restricted the way it is.

## 5.4 Multinomial duopoly

To allow for a more general action set structure, it is possible to allow the agents to open more than one restaurant per period. I now take the same model and change the action sets

to be

$$\mathcal{A}_i(d)\begin{cases}\{1\} & \text{if } d_i = \bar{d}_i \\ \{1,2\} & \text{if } d_i = \bar{d}_1 - 1 \\ \{1,2,3\} & \text{if } d_i \leq \bar{d}_i - 2\end{cases} \tag{78}$$

The corner should then be the same and the two first edge states in either player's direction, but from there on the equilibria might change. The instantaneous profits have not changed from the previous game, but since players can now move faster forwards in the state space more complex competition can arise. It is indeed the case. It's also way out of question to list all equlibria this time around, because there are 85 of them. They are again found using RLS. As in the binary game the diagonal states are attractive, and with new possible actions, there are new ways of getting to them.

$(p_1^1, p_1^2)$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | $(0.00, 1.00)$ | $(0.00, 1.00)$ | $(0.00, 0.00)$ | $(0.00, 0.00)$ |
| 1 | $(1.00, 0.00)$ | $(1.00, 0.00)$ | $(0.00, 1.00)$ | $(0.00, 0.00)$ |
| 2 | $(1.00, 0.00)$ | $(1.00, 0.00)$ | $(0.66, 0.34)$ | $(0.00, 1.00)$ |
| 3 | $-$ | $-$ | $-$ | $-$ |

$(p_2^1, p_2^2)$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | $(0.00, 1.00)$ | $(1.00, 0.00)$ | $(1.00, 0.00)$ | $-$ |
| 1 | $(0.00, 1.00)$ | $(1.00, 0.00)$ | $(1.00, 0.00)$ | $-$ |
| 2 | $(0.00, 0.00)$ | $(0.00, 1.00)$ | $(0.66, 0.34)$ | $-$ |
| 3 | $(0.00, 0.00)$ | $(0.00, 0.00)$ | $(0.00, 1.00)$ | $-$ |

where $p_j^3$ is residually determined. The players move to and stay in $d = (1, 1)$ for a low output high profit strategy if started at $d = (0, 0)$, but if for some reason the model time had started in $d = (2, 1)$ for example, they would move to $d = (2, 2)$ and stay there until one of the mixing actions move them towards the edge or the corner. Comparing the value functions below to the binary duopoly case, we see that it is now possible to acchieve a higher value in a state like $d = (3, 1)$ for player two, because they can now move directly to the corner. We also see that there are significantly more equilibria found in the different states when the

first equilibrium is solved for.

$\mathrm{IV}_1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 349.1 | 349.1 | 344.4 | 341.3 |
| 1 | 349.1 | 349.1 | 344.4 | 340.6 |
| 2 | 344.4 | 344.4 | 344.2 | 340.6 |
| 3 | 341.3 | 340.6 | 340.6 | 340.6 |

$(p_1^1, p_1^2)$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | $(0.0, 1.0)$ | $(0.0, 1.0)$ | $(0.0, 0.0)$ | $(0.0, 0.0)$ |
| 1 | () | 0.7265 | 0.4085 | 0.2351 |
| 2 | 0.9999 | 0.9994 | 0.6273 | 0.6201 |
| 3 | – | – | – | – |

$\mathrm{IV}_2$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 349.1 | 349.1 | 344.4 | 341.3 |
| 1 | 349.1 | 349.1 | 344.4 | 340.6 |
| 2 | 344.4 | 344.4 | 344.1 | 340.6 |
| 3 | 340.3 | 340.6 | 340.6 | 340.6 |

$p_2^1$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | $4 \cdot 10^{-11}$ | 0.9993 | 0.9999 | – |
| 1 | $8 \cdot 10^{-13}$ | 0.7265 | 0.9994 | – |
| 2 | $1 \cdot 10^{-12}$ | 0.4085 | 0.6273 | – |
| 3 | $3 \cdot 10^{-12}$ | 0.2351 | 0.6201 | – |

$n_E Q$ :

| $d_1 \setminus d_2$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 2 | 2 | 1 | 1 |
| 1 | 2 | 3 | 2 | 1 |
| 2 | 1 | 2 | 3 | 1 |
| 3 | 1 | 1 | 1 | 1 |

Of course, it would be all for nothing, if there were no interior states with three actions that had mixed equilibria. The 28th equilibrium found is an example with mixed equilibria in the conditions presented for the multinomial duopoly case. A manually written out example of solving for the equilibrium is found in Appendix G using homotopy continuation and Krawczyk's method. There were four solutions, but only one in the unit box. Krawczyk's method was about five times faster here.

## 5.5 Triopoly game

The equilibria become more unweildly with three players, but I will quickly discuss the results. First, the instantaneous profits has to be adapted to the three player situation. We use the following expression that comes from solving the $\mathcal{N}$-player quantity setter game

$$VP_i = \frac{\bar{x}}{b_1}\left(\frac{b_0 + \sum_{j \neq i} MC_j - \mathcal{N} MC_i}{\mathcal{N}+1}\right)^2 \tag{79}$$

With this expression in hand, we can calculate all quantities needed to setup and solve the equilibrium conditions in the mixed and pure cases. This time we have a corner, edges, interior states where on player is in an absorbing state, and proper interior states. In the proper interior states, there are potentially 16 solutions to the equilibrium conditions. Some of those can be complex, and some of the real solutions might again not be in our domain.

If we look at the sub-stage game played when $d = (1, 1, 1)$ for the 343th equilibrium found, we see that the system has 16 solutions, of which only 8 are real

$$(p_1^1, p_2^1, p_3^1) = (0.141, 0.119, 0.075)$$

In addition to this sub-stage equilibrium, there are three pure strategies: $(p_1^1, p_2^1, p_3^1) = (0.0, 1.0, 1.0)$, $(p_1^1, p_2^1, p_3^1) = (1.0, 0.0, 1.0)$, and $(p_1^1, p_2^1, p_3^1) = (1.0, 1.0, 0.0)$. The third player adds something interesting to the equilibria compared to the other cases. First, so far the diagonal has been the place to agree on a strategy. Since cost structures are equal between players the equilibria had matching strategies on the diagonal.[10] Now, there's a mixed equilibrium with quite different strategy probabilities and the pure strategies in this particular equilibrium has one player staying behind and the others expanding. Again, Appendix G shows the specific example of calculating the mixed sub-stage equilbrium. In this game, a total of 51793 equilibria were found. The homotopy continuation method was 2-3 times faster than Krawczyk here.

# 6 Conclusion

Empirical industrial organization is full of dynamic, strategic interaction. Firms compete through investments, marketing campaigns, price setting, patent races, and more. These models can be hard or impossible to solve analytically. However, for a class of dynamic, directional games we can. As I show in this paper, it is possible to extend the analysis from the binary duopoly to multi-player, multi-action games, even with non-directional states. Both finite and infinite horizon models can be solved, though only the infinite horizon case was shown here.

The equations and solutions given in this paper are straight forward to implement. The homotopy continuation methods for multivariate polynomial systems as well as interval methods and validated numerics were used to solve the complete information games. It will require more benchmarking to find out which method is most appropriate, or in which situations one is preferred over the other. Interval methods successfully solved the binary duopoly games, but more research is needed to come up with a better way of handling the exclusion of edges of the hyperbox that contain no equilibria. The fact that log diverges as the choice probabilities go to zero can simply makes it hard for these methods to converge to a set of unique solutions in reasonable time. It may be possible to come up with better handling of this problem with divergence at the border but that is left for future research to construct more robust and efficient ways of solving incomplete information, directional, dynamic games.

[10]Note, this is of course not a theorem and a proof but only what's been observed.

In this paper I provided clear instructions on how to solve specific model configurations and provided the derivations and steps directlyin the text and appendices. Hopefully, this makes it easier for researchers to derive the exact systems of equations needed for other directional games that might have slightly different structure. The different cases were derived indepently to highlight that special transition structure can make the equilirium conditions more or less complicated to derive and express. The full general model can in principle be expressed explicitly, but at the expense of legibility which itself carries a risk of bugs in implementations and derivations.

# References


Aguirregabiria, V. et al. (2009). Estimation of dynamic discrete games using the nested pseudo likelihood algorithm: code and application. *MPRA Paper 17329*.

Aguirregabiria, V. and P. Mira (2002). Swapping the nested fixed point algorithm: A class of estimators for discrete markov decision models. *Econometrica 70*(4), 1519–1543.

Aguirregabiria, V. and P. Mira (2010). Dynamic discrete choice structural models: A survey. *Journal of Econometrics 156*(1), 38–67.

Barker, K. and C. M. Rocco S (2011). Evaluating uncertainty in risk-based interdependency modeling with interval arithmetic. *Economic Systems Research 23*(2), 213–232.

Bezanson, J., A. Edelman, S. Karpinski, and V. B. Shah (2017). Julia: A fresh approach to numerical computing. *SIAM review 59*(1), 65–98.

Breiding, P. and S. Timme (2018). Homotopycontinuation. jl: A package for homotopy continuation in julia. In *International Congress on Mathematical Software*, pp. 458–465. Springer.

Choobineh, F. and A. Behrens (1993). Use of intervals and possibility distributions in economic analysis. *Journal of the operational research society 43*(9), 907–918.

Datta, R. S. (2010). Finding all nash equilibria of a finite game using polynomial algebra. *Economic Theory 42*(1), 55–96.

Doraszelski, U. and J. F. Escobar (2010). A theory of regular markov perfect equilibria in dynamic stochastic games: Genericity, stability, and purification. *Theoretical Economics 5*(3), 369–402.

Doraszelski, U. and A. Pakes (2007). A framework for applied dynamic analysis in io. *Handbook of industrial organization 3*, 1887–1966.

Dubé, J.-P. H., G. J. Hitsch, and P. K. Chintagunta (2010). Tipping and concentration in markets with indirect network effects. *Marketing Science 29*(2), 216–249.

Fousse, L., G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann (2007). Mpfr: A multiple-precision binary floating-point library with correct rounding. *ACM Transactions on Mathematical Software (TOMS) 33*(2), 13–es.

Hotz, V. J. and R. A. Miller (1993). Conditional choice probabilities and the estimation of dynamic models. *The Review of Economic Studies 60*(3), 497–529.

IEEE (2015). IEEE standard for interval arithmetic. *IEEE Std 1788-2015*, 1–97.

Iskhakov, F., D. Kristensen, J. Rust, and B. Schjerning (2020). Structural estimation of dynamic directional games with multiple equilibria.

Iskhakov, F., J. Rust, and B. Schjerning (2015). Recursive lexicographical search: Finding all markov perfect equilibria of finite state directional dynamic games. *The Review of Economic Studies*, rdv046.

Iskhakov, F., J. Rust, and B. Schjerning (2018). The dynamics of bertrand price competition with cost-reducing investments. *International Economic Review 59*(4), 1681–1731.

Jerrell, M. E. (1997a). Automatic differentiation and interval arithmetic for estimation of disequilibrium models. *Computational Economics 10*(3), 295–316.

Jerrell, M. E. (1997b). Interval arithmetic for input-output models with inexact data. *Computational Economics 10*(1), 89–100.

John, R. (1988). Maximum likelihood estimation of discrete control processes. *SIAM Journal on Control and Optimization 26*(5), 1006–1024.

Judd, K. L., P. Renner, and K. Schmedders (2012). Finding all pure-strategy equilibria in games with continuous strategies. *Quantitative Economics 3*(2), 289–331.

Kubica, B. J. (2019). Interval software, libraries and standards. In *Interval Methods for Solving Nonlinear Constraint Satisfaction, Optimization and Similar Problems*, pp. 91–99. Springer.

Moore, R., R. Kearfott, and M. Cloud (2009). *Introduction To Interval Analysis*. Cambridge Uni Press (CUP).

Revels, J., M. Lubin, and T. Papamarkou (2016). Forward-mode automatic differentiation in Julia. *arXiv:1607.07892 [cs.MS]*.

Revol, N. (2017). Introduction to the ieee 1788-2015 standard for interval arithmetic. In *International Workshop on Numerical Software Verification*, pp. 14–21. Springer.

Rump, S. M. (1980). *Kleine fehlerschranken bei matrixproblemen*.

Rump, S. M. (1999). Intlab—interval laboratory. In *Developments in reliable computing*, pp. 77–104. Springer.

Rust, J. (1987). Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica: Journal of the Econometric Society*, 999–1033.

Rust, J. et al. (1994). Structural estimation of markov decision processes. *Handbook of econometrics 4*(4), 3081–3143.

Toivanen, O., M. Waterson, et al. (2005). Market structure and entry: Where's the beef? *RAND Journal of Economics 36*(3), 680–699.

Verschelde, J. (1999). Algorithm 795: Phcpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software (TOMS) 25*(2), 251–276.

# A The Structure

There is certain structure in our equations that allow us to quite generally derive equilibrium conditions for our models. The structure comes from the multinomial logit structure that allows us to come up with important equations that can be manipulated into equations of unknown choice probabilities $\left\{\{P_i^j(d,x)\}_{j=1}^{J_i(d,x)}\right\}_{i=1}^{\mathcal{N}}$. In principle, I could have solved the conditions in the original $IV_i(d,x)_{i=1}^{\mathcal{N}}$ space, but the advantage of the choice probabilities is that they are bounded between 0 and 1 and the equations that arise from my equation mangling ends up being multivariate polynomials in the complete information case and something close to it in the incomplete information case. So the problem goes from a system of Bellman equations that is not a contraction as a system to a system of bounded variables with specific structure. This is the approach followed in Iskhakov et al. (2015, 2018) for their specific model structure. In this paper I will derive the conditions taking a departure in the integrated value function because I find it very convenient. There are two important

relations I need. In the following, define $\widehat{J}_i \equiv J_i(d,x)$. First, I will use that

$$\begin{aligned} p_i^j(d,x) &= \frac{\exp v_i^j(d,x)/\eta}{\sum_{k=1}^{\widehat{J}} \exp v_i^k(d,x)/\eta} \cdot \frac{\exp v_i^{\widehat{J}}(d,x)/\eta}{\exp v_i^{\widehat{J}}(d,x)/\eta} \\ &= \frac{\exp v_i^{\widehat{J}}(d,x)/\eta}{\sum_{k=1}^{\widehat{J}} \exp v_i^k(d,x)/\eta} \cdot \frac{\exp v_i^j(d,x)/\eta}{\exp v_i^{\widehat{J}}(d,x)/\eta} \\ p_i^j(d,x) &= p_i^{\widehat{J}}(d,x) \frac{\exp v_i^j(d,x)/\eta}{\exp v_i^{\widehat{J}}(d,x)/\eta} \Leftrightarrow \\ \log p_i^j(d,x) - \log p_i^{\widehat{J}}(d,x) &= \frac{v_i^j(d,x)}{\eta} - \frac{v_i^{\widehat{J}}(d,x)}{\eta} \end{aligned} \tag{80}$$

The reason why I expand it using the $\widehat{J}_i$th is that in the model I present this choice will guarantee movement out of the directional part of the state $d$. The unknowns will then be the first $\widehat{J}_i - 1$ choice probabilities for each possible $x$, and apply the simplex restriction to obtain $p_i^{\widehat{J}_i}(d,x)$'s. This means that $v_i^{\widehat{J}_i}(d,x)$ is guaranteed not to include the unknown $IV_i(d,x)$, but only known $IV_i$'s for directional states I've already solved for when $(d,x)$ is reached in the stage recursion. We also note that $v_i^j(d,x)$ may contain $IV_i(d,x)$ but other than that the only things it contain are known $IV_i$'s and unknown choice probabilities for all players for this state, $(d,x)$. Then, I just need to get rid of $IV_i(d,x)$ to reach our goal of only solving equations defined by choice probabilities. To do that I use the fact that

$$\begin{aligned} IV_i(d,x) &= \eta \log \left[ \sum_{j=1}^{\widehat{J}} \exp\left( \frac{v_i^j(d,x)}{\eta} \right) \right] \\ &= \eta \log \left[ \left( \sum_{j=1}^{\widehat{J}} \exp\left( \frac{v_i^j(d,x)}{\eta} \right) \right) \frac{\exp\left( \frac{v_i^J(d,x)}{\eta} \right)}{\exp\left( \frac{v_i^J(d,x)}{\eta} \right)} \right] \\ &= \eta \log \left[ \left( \frac{v_i^{J_i(d,x)}(d,x)}{\eta} \right) \left( \sum_{j=1}^{\widehat{J}} \exp\left( \frac{v_i^j(d,x)}{\eta} \right) \right) \frac{1}{\exp\left( \frac{v_i^J(d,x)}{\eta} \right)} \right] \\ &= \eta \left( \left( \frac{v_i^{J_i(d,x)}(d,x)}{\eta} \right) + \log \left[ \left( \sum_{j=1}^{\widehat{J}} \exp\left( \frac{v_i^j(d,x)}{\eta} \right) \right) \frac{1}{\exp\left( \frac{v_i^J(d,x)}{\eta} \right)} \right] \right) \\ &= \eta \left( \left( \frac{v_i^{J_i(d,x)}(d,x)}{\eta} \right) + \log \left[ \left( p_i^{J_i(d,x)}(d,x) \right)^{-1} \right] \right) \\ &= v_i^{J_i(d,x)}(d,x) - \eta \log \left[ p_i^{J_i(d,x)}(d,x) \right] \end{aligned} \tag{81}$$

With these definitons in hand I can take the first $J_i(d,x) - 1$ choice probability equations for each player and use them as the equilibrium conditions. We write them in the eq. (80) form and apply eq. (81) where needed.

# B Equilibrium conditions for the binary duopoly

Here I show how to derive the alternative equilibrium conditions for the binary duopoly game. Let us see how. We have

$$\begin{aligned}
v_1^1(d) &= p_2^1(d)\Pi_1(d,1,1) + p_2^2(d)\Pi_1(d,1,2) + \beta_1\left(p_2^1(d)\mathrm{IV}_1(d) + p_2^2(d)\mathrm{IV}_1((d_1,d_2+1))\right) \\
&= \tau_1(d,1,2) + p_2^1(d)\left[\Pi_1(d,1,1) - \tau_1(d,1,2)\right] + p_2^1(d)\beta_1\mathrm{IV}_1(d) \\
v_1^2(d) &= p_2^1(d)\Pi_1(d,2,1) + p_2^2(d)\Pi_1(d,2,2) + \beta_1\left(p_2^1(d)\mathrm{IV}_1((d_1+1,d_2)) + p_2^2(d)\mathrm{IV}_1(d+1)\right) \\
&= \tau_1(d,2,2) + p_2^1(d)\left[\tau_1(d,2,1) - \tau_1(d,2,2)\right] \qquad (82)
\end{aligned}$$

as well as eq. (80). This gives us

$$\begin{aligned}
0 &= v_1^1(d) - \eta\log\left(\frac{p_1^1(d)}{p_1^2(d)}\right) - v_1^2(d) \\
&= K_1^1 - \eta\log\left(\frac{p_1^1(d)}{p_1^2(d)}\right) + \tilde{K}_1^2 p_2^1(d) + p_2^2(d)\beta_1\mathrm{IV}_1(d) \qquad (83) \\
K_1^1 &\equiv \left[\tau_1(d,1,2) - \tau_1(d,2,2)\right] \qquad (84) \\
\tilde{K}_1^2 &\equiv \Big[\Pi_1(d,1,1) + \tau_1(d,2,2) - \tau_1(d,2,1) - \tau_1(d,1,2)\Big] \qquad (85)
\end{aligned}$$

As $p_2^1(s)$ enters $v_1^1(d)$ linearly (remember $p_2^2(d) = 1 - p_2^1$), and all unknowns but $p_2^1$ and $p_2^2$ have been eliminated, it should be obvious by now that it's heading towards a system of two equations in two unknowns, and, looking at eq. (83), that they are going to be something close to a system of polynomials. Looking at eq. (82) and eq. (81) I get the following

$$p_2^1(d)\beta_1\mathrm{IV}_1(d) = p_2^1(d)\beta_1\left(\tau_1(d,2,2) + p_2^1(d)\left[\tau_1(d,2,1) - \tau_1(d,2,2)\right] - \eta\log\left[p_1^2(d)\right]\right)$$

substituting this back into eq. (83) I get

$$\begin{aligned}
0 &= K_1^1 - \eta\log\left(\frac{p_1^1(d)}{1-p_1^1(d)}\right) + \left(K_1^2 - \beta_1\eta\log\left[1-p_1^1(d)\right]\right)p_2^1(d) + K_1^3 p_2^2(d)^2 \qquad (86) \\
K_1^1 &\equiv \left[\tau_1(d,1,2) - \tau_1(d,2,2)\right] \\
K_1^2 &\equiv \Big[\Pi_1(d,1,1) + \tau_1(d,2,2) - \tau_1(d,2,1) - \tau_1(d,1,2)\Big] + \beta_1\tau_1(d,2,2) \\
K_1^3 &\equiv \beta_1\left(\tau_1(d,2,1) - \tau_1(d,2,2)\right)
\end{aligned}$$

and starting from player 2 value functions I get

$$\begin{aligned}
0 &= K_2^1 - \eta\log\left(\frac{p_2^1(d)}{1-p_2^1(d)}\right) + \left(K_2^2 - \beta_2\eta\log\left[1-p_2^1(d)\right]\right)p_1^1(d) + K_2^3 p_1^2(d)^2 \qquad (87) \\
K_2^1 &\equiv \left[\tau_2(d,2,1) - \tau_2(d,2,2)\right] \\
K_2^2 &\equiv \Big[\Pi_2(d,1,1) + \tau_2(d,2,2) - \tau_2(d,1,2) - \tau_2(d,2,1)\Big] + \beta_2\tau_2(d,2,2) \\
K_2^3 &\equiv \beta_2\left(\tau_2(d,1,2) - \tau_2(d,2,2)\right)
\end{aligned}$$

This completes the derivation of the equilibrium conditions in the binomial duopoly.

# C Equilibrium conditions for the multinomial duopoly

In this appendix I derive the multinomial choice case. Again, I start from $v_1^j(d)$ and relate it to $v_1^{\widehat{J}_1}(d)$ where $\widehat{J}_i \equiv J_i(d)$. Then

$$\begin{aligned} v_1^1(d) &= p_2^1(d)\Pi_1(d,1,1) + \sum_{k=2}^{\widehat{J}_2-1} p_2^k \Pi(d,1,k) + (1 - p_2^{-\widehat{J}_2}(d))\Pi_1(d,1,\widehat{J}_2) \\ &+ \beta_1 \Big[ p_2^1(d) IV_1(d) + \sum_{k=2}^{\widehat{J}_2-1} p_2^k IV_1((d_1, d_2 + k - 1) \\ &+ \left(1 - p_2^{-\widehat{J}_2}(d)\right) IV_1((d_1, d_2 + \widehat{J}_2 - 1)) \Big] \\ &= \tau_i(d,1,\widehat{J}_2) + p_2^1(d)\left(\Pi_1(d,1,1) - \tau_1(d,1,\widehat{J}_2)\right) + \beta_1 p_2^1(d) IV_1(d) \\ &\qquad + \sum_{k=2}^{\widehat{J}_2-1} p_2^k \left(\tau_1(d,1,k) - \tau_1(d,1,\widehat{J}_2)\right) \end{aligned} \tag{88}$$

and for $1 < j \leq J_1(d)$

$$\begin{aligned} v_1^j(d) &= \sum_{k=1}^{\widehat{J}_2-1} p_2^k(d)\Pi(d,j,k) + (1 - p_2^{-\widehat{J}_2}(d))\Pi_1(d,j,\widehat{J}_2) \\ &+ \beta_1 \Big[ \sum_{k=1}^{\widehat{J}_2-1} p_2^k(d) IV_1((d_1 + j - 1, d_2 + k - 1) \\ &+ \left(1 - p_2^{-\widehat{J}_2}(d)\right) IV_1((d_1 + j - 1, d_2 + \widehat{J}_2 - 1)) \Big] \qquad (89) \\ &= \tau_1(d,j,\widehat{J}_2) + \sum_{k=1}^{\widehat{J}_2-1} p_2^k(d)\left(\tau_1(d,j,k) - \tau_1(d,j,\widehat{J}_2)\right) \qquad (90) \end{aligned}$$

where eq. (88) is the same as in the binary duopoly case if $\widehat{J}_1(d) = 2$. First, simplify the first equation

$$
\begin{aligned}
0 &= v_1^1(d) - v_1^{\widehat{J}}(d) - \eta \log\left(\frac{p_1^1(d)}{p_1^{\widehat{J}_1}(d)}\right) \qquad (91)\\
&= \Big[\tau_i(d,1,\widehat{J}_2) + p_2^1(d)\left(\Pi_1(d,1,1) - \tau_1(d,1,\widehat{J}_2)\right) + \beta_1 p_2^1(d) IV_1(d) \\
&\quad + \sum_{k=2}^{\widehat{J}_2-1} p_2^k\left(\tau_1(d,1,k) - \tau_1(d,1,\widehat{J}_2)\right)\Big] \\
&\quad - \Big[\tau_1(d,\widehat{J}_1,\widehat{J}_2) + \sum_{k=1}^{\widehat{J}_2-1} p_2^k(d)\left(\tau_1(d,\widehat{J}_1,k) - \tau_1(d,\widehat{J}_1,\widehat{J}_2)\right)\Big] \\
&\quad - \eta \log\left(\frac{p_1^1(d)}{p_1^{\widehat{J}_1}(d)}\right) \qquad (92)
\end{aligned}
$$

from which I get the first equilibrium equation for player 1

$$
\begin{aligned}
0 &= L_1^1 - \eta \log\left(\frac{p_1^1(d)}{p_1^{\widehat{J}_1}(d)}\right) + \left(L_1^2 - \beta_1 \eta \log\left(p_1^{\widehat{J}_1}\right)\right) p_2^1(d) \\
&\quad + p_2^1(d) \sum_{k=1}^{\widehat{J}_2-1} L_1^{3,k} p_2^k(d) + \sum_{k=2}^{\widehat{J}_2-1} L_1^{4,k} p_2^k(d) \qquad (93)\\
L_1^1 &\equiv \left[\tau_1(d,1,\widehat{J}_2) - \tau_1(d,\widehat{J}_1,\widehat{J}_2)\right] \qquad (94)\\
L_1^2 &\equiv \left[\Pi_1(d,1,1) - \tau_1(d,1,\widehat{J}_2) - \tau_1(d,\widehat{J}_1,1) + \tau_1(d,\widehat{J}_1,\widehat{J}_2)\right] + \beta_1 \tau_1(d,\widehat{J}_1,\widehat{J}_2) \qquad (95)\\
L_1^{3,k} &\equiv \beta_1\left[\tau_1(d,\widehat{J}_1,k) - \tau_1(d,\widehat{J}_1,\widehat{J}_2)\right] \qquad (96)\\
L_1^{4,k} &\equiv \left[\tau_1(d,1,k) - \tau_1(d,1,\widehat{J}_2) - \tau_1(d,\widehat{J}_1,k) + \tau_1(d,\widehat{J}_1,\widehat{J}_2)\right] \qquad (97)
\end{aligned}
$$

which again is the binary solution with an added term. The player 2 solution is

$$
\begin{aligned}
0 &= L_2^1 - \eta \log\left(\frac{p_2^1(d)}{p_2^{\widehat{J}_2}(d)}\right) + \left(L_2^2 - \beta_2 \eta \log\left(p_2^{\widehat{J}_2}\right)\right) p_1^1(d) \\
&\quad + p_1^1(d) \sum_{k=1}^{\widehat{J}_1-1} L_2^{3,k} p_1^k(d) + \sum_{k=2}^{\widehat{J}_1-1} L_2^{4,k} p_1^k(d) \qquad (98)\\
L_2^1 &\equiv \left[\tau_2(d,\widehat{J}_1,1) - \tau_2(d,\widehat{J}_1,\widehat{J}_2)\right] \qquad (99)\\
L_2^2 &\equiv \left[\Pi_2(d,1,1) - \tau_2(d,\widehat{J}_1,1) - \tau_2(d,1,\widehat{J}_2) + \tau_2(d,\widehat{J}_1,\widehat{J}_2)\right] + \beta_2 \tau_2(d,\widehat{J}_1,\widehat{J}_2) \qquad (100)\\
L_2^{3,k} &\equiv \beta_2\left[\tau_2(d,k,\widehat{J}_2) - \tau_2(d,\widehat{J}_1,\widehat{J}_2)\right] \qquad (101)\\
L_2^{4,k} &\equiv \left[\tau_2(d,k,1) - \tau_2(d,\widehat{J}_1,1) - \tau_2(d,k,\widehat{J}_2) + \tau_2(d,\widehat{J}_1,\widehat{J}_2)\right] \qquad (102)
\end{aligned}
$$

The equation for the remaining equilibrium conditions are given by

$$0 = v_1^j(d) - v_1^{\widehat{J}}(d) - \eta \log\left(\frac{p_1^j(d)}{p_1^{\widehat{J}_1}(d)}\right) \tag{103}$$

$$\begin{aligned} = \quad & \left[\tau_1(d,j,\widehat{J}_2) + \sum_{k=1}^{\widehat{J}_2-1} p_2^k(d)\left(\tau_1(d,j,k) - \tau_1(d,j,\widehat{J}_2)\right)\right] \\ & - \left[\tau_1(d,\widehat{J}_1,\widehat{J}_2) + \sum_{k=1}^{\widehat{J}_2-1} p_2^k(d)\left(\tau_1(d,\widehat{J}_1,k) - \tau_1(d,\widehat{J}_1,\widehat{J}_2)\right)\right] \\ & - \eta \log\left(\frac{p_1^j(d)}{p_1^{\widehat{J}_1}(d)}\right) \end{aligned} \tag{104}$$

from which I get the simpler equations for $j_1 \in 2, \ldots, J(d)_1 - 1$

$$0 = L_1^5 - \eta \log\left(\frac{p_1^{j_1}(d)}{p_1^{\widehat{J}_1}(d)}\right) + \sum_{k=1}^{\widehat{J}_2-1} L_1^{6,k} p_2^k(d) \tag{105}$$

$$L_1^5 \equiv \left[\tau_1(d,j_1,\widehat{J}_2) - \tau_1(d,\widehat{J}_1,\widehat{J}_2)\right] \tag{106}$$

$$L_1^{6,k} \equiv \left[\tau_1(d,j_1,k) - \tau_1(d,j_1,\widehat{J}_2) - \tau_1(d,\widehat{J}_1,k) + \tau_1(d,\widehat{J}_1,\widehat{J}_2)\right] \tag{107}$$

These are simpler than the first one, because as soon as player one choses an action indexed larger than 1, it is impossible to stay in the current state next period. For completeness, the equations for player 2 for $j_2 \in 2, \ldots, J(d)_2 - 1$ are

$$0 = L_2^5 - \eta \log\left(\frac{p_2^{j_2}(d)}{p_2^{\widehat{J}_2}(d)}\right) + \sum_{k=1}^{\widehat{J}_2-1} L_2^{6,k} p_1^k(d) \tag{108}$$

$$L_2^5 \equiv \left[\tau_1(d,\widehat{J}_1,j_2) - \tau_2(d,\widehat{J}_1,\widehat{J}_2)\right] \tag{109}$$

$$L_2^{6,k} \equiv \left[\tau_2(d,k,j_2) - \tau_2(d,\widehat{J}_1,j_2) - \tau_2(d,k,\widehat{J}_2) + \tau_2(d,\widehat{J}_1,\widehat{J}_2)\right] \tag{110}$$

This completes the derivation of the equilibrium conditions in the multinomial duopoly.

# D Equilibrium conditions for the binary triopoly

As above, I start from the individual choice specific value functions because they enter the re-written equilirium conditions.

$$
\begin{aligned}
v_1^1(d) &= p_2^1(d)p_3^1(d)\Pi_1(d,1,1,1) + p_2^1(d)p_3^1(d)\beta_1 IV(d) + p_2^1(d)(1-p_3^1(d))\tau_1(d,1,1,2) && (111)\\
&\quad + (1-p_2^1(d))p_3^1(d)\tau_1(d,1,2,1) + (1-p_2^1(d))(1-p_3^1(d))\tau_1(d,1,2,2) && (112)\\
&= p_2^1(d)p_3^1(d)\left[\Pi_1(d,1,1,1) - \tau_1(d,1,1,2) - \tau_1(d,1,2,1) - \tau_1(d,1,2,2)\right] && (113)\\
&\quad + p_2^1(d)\left[\tau_1(d,1,1,2) - \tau_1(d,1,2,2)\right] + p_3^1(d)\left[\tau_1(d,1,2,1) - \tau_1(d,1,2,2)\right] && (114)\\
&\quad + \tau_1(d,1,2,2) + p_2^1(d)p_3^1(d)\beta_1 IV(d) && (115)
\end{aligned}
$$

$$
\begin{aligned}
v_1^2(d) &= p_2^1(d)p_3^1(d)\tau_1(d,2,1,1) + p_2^1(d)(1-p_3^1(d))\tau_1(d,2,1,2) && (116)\\
&\quad + (1-p_2^1(d))p_3^1(d)\tau_1(d,2,2,1) + (1-p_2^1(d))(1-p_3^1(d))\tau_1(d,2,2,2) && (117)\\
&= p_2^1(d)p_3^1(d)\left[\tau_1(d,2,1,1) - \tau_1(d,2,1,2) - \tau_1(d,2,2,1) - \tau_1(d,2,2,2)\right] && (118)\\
&\quad + p_2^1(d)\left[\tau_1(d,2,1,2) - \tau_1(d,2,2,2)\right] + p_3^1(d)\left[\tau_1(d,2,2,1) - \tau_1(d,2,2,2)\right] && (119)\\
&\quad + \tau_1(d,2,2,2) && (120)
\end{aligned}
$$

And using eq. (80)

$$
\begin{aligned}
0 &= v_1^1(d) - v_1^2(d) - \eta\log\left[\frac{p_1^1(d)}{1-p_1^1(d)}\right]\\
&= M_1^1 - \eta\log\left[\frac{p_1^1(d)}{1-p_1^2(d)}\right] + p_2^1(d)M_1^2 + p_3^1(d)M_1^3 + p_2^1(d)p_3^1(d)M_1^4 + p_2^1(d)p_3^1(d)\beta_1 IV(d)\\
M_1^1 &\equiv \left[\tau_1(d,1,2,2) - \tau_1(d,2,2,2)\right]\\
M_1^2 &\equiv \left[\tau_1(d,1,1,2) - \tau_1(d,1,2,2) - \tau_1(d,2,1,2) + \tau_1(d,2,2,2)\right]\\
M_1^3 &\equiv \left[\tau_1(d,1,2,1) - \tau_1(d,1,2,2) - \tau_1(d,2,2,1) + \tau_1(d,2,2,2)\right]\\
\tilde{M}_1^4 &\equiv \Big[\Pi_1(d,1,1,1) - \tau_1(d,1,1,2) - \tau_1(d,1,2,1) - \tau_1(d,1,2,2)\\
&\qquad - \tau_1(d,2,1,1) + \tau_1(d,2,1,2) + \tau_1(d,2,2,1) + \tau_1(d,2,2,2)\Big]
\end{aligned}
$$

and using eq. (81) I get

$$
\begin{aligned}
0 = M_1^1 - \eta \log\left[\frac{p_1^1(d)}{1-p_1^1(d)}\right] + M_1^2 p_2^1(d) + M_1^3 p_3^1(d) + \left[M_1^4 - \beta_1 \eta \log\left[p_1^2(d)\right]\right] p_2^1(d) p_3^1(d) \\
+ M_1^5 p_2^1(d)^2 p_3^1(d) + M_1^6 p_2^1(d) p_3^1(d)^2 + M_1^7 p_2^1(d)^2 p_3^1(d)^2
\end{aligned}
$$

$$
\begin{aligned}
M_1^1 &\equiv [\tau_1(d,1,2,2) - \tau_1(d,2,2,2)] \\
M_1^2 &\equiv [\tau_1(d,1,1,2) - \tau_1(d,1,2,2) - \tau_1(d,2,1,2) + \tau_1(d,2,2,2)] \\
M_1^3 &\equiv [\tau_1(d,1,2,1) - \tau_1(d,1,2,2) - \tau_1(d,2,2,1) + \tau_1(d,2,2,2)] \\
M_1^4 &\equiv \Big[\Pi_1(d,1,1,1) - \tau_1(d,1,1,2) - \tau_1(d,1,2,1) - \tau_1(d,1,2,2) \\
&\qquad - \tau_1(d,2,1,1) + \tau_1(d,2,1,2) + \tau_1(d,2,2,1) + (1+\beta_1)\tau_1(d,2,2,2)\Big] \\
M_1^5 &\equiv \beta_1 [\tau_1(d,2,1,2) - \tau_1(d,2,2,2)] \\
M_1^6 &\equiv \beta_1 [\tau_1(d,2,2,1) - \tau_1(d,2,2,2)] \\
M_1^7 &\equiv \beta_1 [\tau_1(d,2,1,1) - \tau_1(d,2,1,2) - \tau_1(d,2,2,1) - \tau_1(d,2,2,2)]
\end{aligned}
$$

and for player 2

$$
\begin{aligned}
0 = M_2^1 - \eta \log\left[\frac{p_2^1(d)}{1-p_2^1(d)}\right] + M_2^2 p_1^1(d) + M_2^3 p_3^1(d) + \left[M_2^4 - \beta_1 \eta \log\left[p_2^2(d)\right]\right] p_1^1(d) p_3^1(d) \\
+ M_2^5 p_1^1(d)^2 p_3^1(d) + M_2^6 p_1^1(d) p_3^1(d)^2 + M_2^7 p_1^1(d)^2 p_3^1(d)^2
\end{aligned}
$$

$$
\begin{aligned}
M_2^1 &\equiv [\tau_2(d,2,1,2) - \tau_2(d,2,2,2)] \\
M_2^2 &\equiv [\tau_2(d,1,1,2) - \tau_2(d,2,1,2) - \tau_2(d,1,2,2) + \tau_2(d,2,2,2)] \\
M_2^3 &\equiv [\tau_2(d,2,1,1) - \tau_2(d,2,1,2) - \tau_2(d,2,2,1) + \tau_2(d,2,2,2)] \\
M_2^4 &\equiv \Big[\Pi_2(d,1,1,1) - \tau_2(d,1,1,2) - \tau_2(d,2,1,1) - \tau_2(d,2,1,2) \\
&\qquad - \tau_2(d,1,2,1) + \tau_2(d,1,2,2) + \tau_2(d,2,2,1) + (1+\beta_2)\tau_2(d,2,2,2)\Big] \\
M_2^5 &\equiv \beta_2 [\tau_2(d,1,2,2) - \tau_2(d,2,2,2)] \\
M_2^6 &\equiv \beta_2 [\tau_2(d,2,2,1) - \tau_2(d,2,2,2)] \\
M_2^7 &\equiv \beta_2 [\tau_2(d,1,2,1) - \tau_2(d,1,2,2) - \tau_2(d,2,2,1) - \tau_2(d,2,2,2)]
\end{aligned}
$$

and for player 3

$$
\begin{aligned}
0 = M_3^1 - \eta \log\left[\frac{p_3^1(d)}{1-p_3^2(d)}\right] + M_1^2 p_2^1(d) + M_3^3 p_1^1(d) + \left[M_3^4 - \beta_3 \eta \log\left[p_3^2(d)\right]\right] p_1^1(d) p_2^1(d) \\
+ M_3^5 p_2^1(d)^2 p_1^1(d) + M_3^6 p_2^1(d) p_1^1(d)^2 + M_3^7 p_2^1(d)^2 p_1^1(d)^2
\end{aligned}
$$

$$
\begin{aligned}
M_3^1 &\equiv [\tau_3(d,2,2,1) - \tau_3(d,2,2,2)] \\
M_3^2 &\equiv [\tau_3(d,2,1,1) - \tau_3(d,2,2,1) - \tau_3(d,2,1,2) + \tau_3(d,2,2,2)] \\
M_3^3 &\equiv [\tau_3(d,1,2,1) - \tau_3(d,2,2,1) - \tau_3(d,1,2,2) + \tau_3(d,2,2,2)] \\
M_3^4 &\equiv \Big[\Pi_3(d,1,1,1) - \tau_3(d,2,1,1) - \tau_3(d,1,2,1) - \tau_3(d,2,2,1) \\
&\qquad - \tau_3(d,1,1,2) + \tau_3(d,2,1,2) + \tau_3(d,1,2,2) + (1+\beta_3)\tau_3(d,2,2,2)\Big] \\
M_3^5 &\equiv \beta_3 [\tau_3(d,2,1,2) - \tau_3(d,2,2,2)] \\
M_3^6 &\equiv \beta_3 [\tau_3(d,1,2,2) - \tau_3(d,2,2,2)] \\
M_3^7 &\equiv \beta_3 [\tau_3(d,1,1,2) - \tau_3(d,2,1,2) - \tau_3(d,1,2,2) - \tau_3(d,2,2,2)]
\end{aligned}
$$

This completes the derivation of the equilibrium conditions in the binomial triopoly and it's still the multivariate polynomial system case in the complete information game.

# E Equilibrium conditions for binary duopoly with exogenous non-directional states

For the model with exogenous non-directional states there is a non-empty $\mathcal{X}$. Once again, I need to set up the equilibrium conditions from the re-written choice probabilities. We still get equations for the first $J_i(d,x)-1$ choices, but since I need to solve for all non-directional states at once each sub-stages will have $(J_i(d,x)-1) \times |\mathcal{X}|$ equations and unknowns. For

$x \in \mathcal{X}(d)$ we get

$$
\begin{aligned}
0 &= v_1^1(d,x) - v_1^2(d,x) - \eta \frac{\log\left[p_1^1(d,x)\right]}{\log\left[p_1^2(d,x)\right]} \\
&= \left[ p_2^1(d,x) \left[ \Pi_1(d,x,1,1) + \beta_1 \sum_{x' \in \mathcal{X}} \pi^{\mathcal{X}}(x'|x) IV(d,x',1,1) \right] + (1 - p_2^1(d,x)) \tau_1(d,x,1,2) \right] \\
&\quad - \left[ p_2^1(d,x) \left[ \tau(d,x,2,1) - \tau(d,x,2,2) \right] + \tau(d,x,2,2) \right] \\
&\quad - \eta \frac{\log\left[p_1^1(d,x)\right]}{\log\left[p_1^2(d,x)\right]} \\
&= Q_1^1 - \eta \frac{\log\left[p_1^1(d,x)\right]}{\log\left[p_1^2(d,x)\right]} + \tilde{Q}_1^2 p_2^1(d,x) + \left[ p_2^1(d,x) \left[ \beta_1 \sum_{x' \in \mathcal{X}} \pi^{\mathcal{X}}(x'|x) IV(d,x',1,1) \right] \right]
\end{aligned}
$$

$$
\begin{aligned}
Q_1^1 &\equiv \left[ \tau(d,x,1,2) - \tau_1(d,x,2,2) \right] \\
\tilde{Q}_1^2 &\equiv \left[ \Pi_1(d,x,1,1) + \tau_1(d,x,2,2) - \tau_1(d,x,1,2) - \tau_1(d,x,2,1) \right]
\end{aligned}
$$

and apply eq. (81)

$$
\begin{aligned}
0 &= Q_1^1 - \eta \frac{\log\left[p_1^1(d,x)\right]}{1 - \log\left[p_1^1(d,x)\right]} + \tilde{Q}_1^2 p_2^1(d,x) + \left[ p_2^1(d,x) \left[ \beta_1 \sum_{x' \in \mathcal{X}} \pi^{\mathcal{X}}(x'|x) IV(d,x',1,1) \right] \right] \\
&= Q_1^1 - \eta \frac{\log\left[p_1^1(d,x)\right]}{\log\left[1 - p_1^1(d,x)\right]} + \left[ \tilde{Q}_1^2 - \beta_1 \eta \sum_{x' \in \mathcal{X}} \pi^{\mathcal{X}}(x'|x) \log\left[p_1^2(d,x')\right] \right] p_2^1(d,x) \\
&\quad + \left[ p_2^1(d,x) \left[ \beta_1 \sum_{x' \in \mathcal{X}} \pi^{\mathcal{X}}(x'|x) \left[ p_2^1(d,x') (\tau_1(d,x',2,1) - \tau_1(d,x',2,2)) + \tau_1(d,x',2,2) \right] \right] \right] \\
&= Q_1^1 - \eta \frac{\log\left[p_1^1(d,x)\right]}{1 - \log\left[p_1^1(d,x)\right]} + \left[ Q_1^2 - \beta_1 \eta \sum_{x' \in \mathcal{X}} \pi^{\mathcal{X}}(x'|x) \log\left[p_1^2(d,x')\right] \right] p_2^1(d,x) \\
&\quad + p_2^1(d,x) \sum_{x' \in \mathcal{X}} p_2^1(d,x') Q_1^{3,j}
\end{aligned}
$$

$$
\begin{aligned}
Q_1^1 &\equiv \left[ \tau(d,x,1,2) - \tau_1(d,x,2,2) \right] \\
Q_1^2 &\equiv \left[ \Pi_1(d,x,1,1) + \tau_1(d,x,2,2) - \tau_1(d,x,1,2) - \tau_1(d,x,2,1) \right] \\
&\qquad + \beta_1 \sum_{x' \in \mathcal{X}} \pi^{\mathcal{X}}(x'|x) \tau_1(d,x',2,2) \\
Q_1^{3,j} &\equiv \beta_1 \pi^{\mathcal{X}}(x'|x) (\tau_1(d,x',2,1) - \tau_1(d,x',2,2))
\end{aligned}
$$

and likewise for player 2

$$
\begin{aligned}
0 = Q_2^1 - \eta \frac{\log\left[p_2^1(d,x)\right]}{1-\log\left[p_2^1(d,x)\right]} + \left[Q_2^2 - \beta_2 \eta \sum_{x' \in \mathcal{X}} \pi^{\mathcal{X}}(x'|x) \log\left[p_2^2(d,x')\right]\right] p_1^1(d,x) \\
+ p_1^1(d,x) \sum_{x' \in \mathcal{X}} p_1^1(d,x') Q_2^{3,j} \\
Q_2^1 \equiv \left[\tau(d,x,2,1) - \tau_2(d,x,2,2)\right] \\
Q_2^2 \equiv \Big[\Pi_2(d,x,1,1) + \tau_2(d,x,2,2) - \tau_2(d,x,2,1) - \tau_2(d,x,1,2)\Big] \\
+ \beta_2 \sum_{x' \in \mathcal{X}} \pi^{\mathcal{X}}(x'|x) \tau_2(d,x',2,2) \\
Q_2^{3,j} \equiv \beta_2 \pi^{\mathcal{X}}(x'|x)(\tau_2(d,x',1,2) - \tau_2(d,x',2,2))
\end{aligned}
$$

This completes the derivation of the equilibrium conditions in the binomial duopoly with exogenous states.

# F Equilibrium conditions for a model with actions as distributions over future states

If the deterministic transitions used in this paper's example is not sufficiently flexible to model some strategic interaction, it might be necessary to allow for stochastic transitions. In the main text an example is mentioned where the $d_i + a_i$ transition is inteded from the agent's side but there are outside forces that might make the expansion fail. Then some probabilities have to be assigned to the different transitions $d_i \to d_i'$ or maybe even the full vector of $d'$'s. Here, I will explain why such models will also exhibit the same structure in the equilibrium conditions as the simpler models.

The model structure with polynomial terms of choice probabilities with some coefficients including log's comes from separating out the $IV_i(d)$ terms in the choice specific value functions as shown in Appendix A and any of the game specific appendices. In this more complicated setup, the structure from Appendix A still apply, so the question is if there's anything in the choice specific value functions that might get in our way.

$$
v_i^j(d,x) = \sum_{d' \in \mathcal{D}} \left[ \sum_{a \in \mathcal{A}} \left( \prod_{k \neq i} \sigma_k^{a_k} \pi^{\mathcal{A}}(d'|d,a) \right) \left[ u_i(d,x,a) + \beta_i \sum_{x' \in \mathcal{X}} IV_i(d',x') \pi^{\mathcal{X}}(x'|x) \right] \right]
$$

Now, written like this, it's quite obvious what needs to be done. Take the current directional

state $d$ outside of the first sum.

$$v_i^j(d,x) = \sum_{a\in\mathcal{A}} \left( \prod_{k\neq i} \sigma_k^{a_k} \pi^{\mathcal{A}}(d|d,a) \right) \left[ u_i(d,x,a) + \beta_i \sum_{x'\in\mathcal{X}} IV_i(d,x')\pi^{\mathcal{X}}(x'|x) \right] + \sum_{d'\in\mathcal{D}\setminus d} \left[ \sum_{a\in\mathcal{A}} \left( \prod_{k\neq i} \sigma_k^{a_k} \pi^{\mathcal{A}}(d'|d,a) \right) \left[ u_i(d,x,a) + \beta_i \sum_{x'\in\mathcal{X}} IV_i(d',x')\pi^{\mathcal{X}}(x'|x) \right] \right]$$

Using the substitution where we replace for $\sigma_k^{J_k(d)}$'s with one minus the rest of the strategy probabilities and substituting in for $IV(d,x)$ we will again get multiplications of the exact same kind as in the examples in this paper. However, if many choices can lead to $d$ and if these transition rules are not sparse, then there are clearly many terms to multiply together, and the result would be polynomials of quite high order. It's also clear that at least one choice has to imply a certain movement in the directional state or else the current value function cannot be substituted away.

# G Code snippets for mixed interior equilibria

## G.1 Binary duopoly

In the binary duopoly game there can be mixed equilibria in the interior states. In the main text I refer to this appendix for the example of solving for mixed equilibria in the $d = (2,2)$ state. A written out example of that, taking the constants given in Appendix B for given, is

```
using HomotopyContinuation

K11, K12, K13 = -0.2748148148148175, -0.02370370370368846, 0.6699259259259406
K21, K22, K23 = -0.2748148148148175, -0.02370370370368846, 0.6699259259259406)

@var p1, p2
F = System([K11 + K12*p2 + K13*p2^2,
            K21 + K22*p1 + K23*p1^2],
            variables = [p1, p2])
sol = solve(F)
```

which prints

```
    Result with 4 solutions
    =======================
```

- 4 paths tracked
- 4 non-singular solutions (4 real)
- random_seed: 0x58a0964e
- start_system: :polyhedral

the real solutions can be grabbed with

```
julia> real_solutions(sol)
4-element Array{Array{Float64,1},1}:
 [0.6584172034074091, 0.6584172034074091]
 [0.6584172034074091, -0.6230346293251682]
 [-0.6230346293251682, 0.6584172034074091]
 [-0.6230346293251682, -0.6230346293251682]
```

which only have one feasible solution to our bounded problem. We can compare this to the solutions found using interval arithmetic.

```
using IntervalRootFinding, IntervalArithmetic
using StaticArrays
function f(Ps)
    P1, P2 = Ps

    K11, K12, K13 = -0.2748148148148175, -0.02370370370368846, 0.6699259259259406
    K21, K22, K23 = -0.2748148148148175, -0.02370370370368846, 0.6699259259259406

    return SVector{2}([K11 + K12*P2 + K13*P2^2,
                       K21 + K22*P1 + K23*P1^2])
end

int01 = interval(0.0, 1.0)

rts = roots(f, int01 × int01, Krawczyk, 1e-12)
```

Which prints

```
1-element Array{Root{IntervalBox{2,Float64}},1}:
 Root([0.658417, 0.658418] × [0.658417, 0.658418], :unique)
```

So the two methods agree, and the interval method which is well-defined on the full box in the complete information game, and it does verify that the solution is indeed unique in the

unit box. The intervals look large, but the printing is not true to the specific values. It rounds up to show that the values are different. We can query the exact bounds

```
julia> rts[1].interval[1].lo
0.6584172034074088

julia> rts[1].interval[1].hi
0.6584172034074093
```

As described in section 3, the status of ":unique" means that it's mathematically sure, given the functions involved in the system of equations, that there is only one unique root between those two floating point values.

## G.2 Multinomial duopoly

```
julia> K11, K12, K13k, K14k = -0.9022222222222354, -0.09481481481481069,
[1.3511111111111063, 0.6699259259259406], [5.6183131848704875,
-0.047407407407433766]
(-0.9022222222222354, -0.09481481481481069, [1.3511111111111063,
0.6699259259259406],
[5.6183131848704875, -0.047407407407433766])

julia> K21, K22, K23k, K24k = -0.9022222222222354, -0.09481481481481069,
[1.3511111111111063, 0.6699259259259406], [5.6183131848704875,
-0.047407407407433766]
(-0.9022222222222354, -0.09481481481481069, [1.3511111111111063,
0.6699259259259406], [5.6183131848704875, -0.047407407407433766])

julia> K15, K16k = -0.2748148148148175, [-0.04740740740737692,
3.417860807145871]
(-0.2748148148148175, [-0.04740740740737692, 3.417860807145871])

julia> K25, K26k = -0.2748148148148175, [-0.04740740740737692,
3.417860807145871]
(-0.2748148148148175, [-0.04740740740737692, 3.417860807145871])

julia> @var p11, p12, p21, p22
```

```
(p11, p12, p21, p22)

julia> F = System([K11 + K12*p21 + K13k[1]*p21^2 + K13k[2]*p21*p22
+ K14k[2]*p22,
                   K21 + K22*p11 + K23k[1]*p11^2 + K23k[2]*p11*p12
+ K24k[2]*p12,
                   K15 + K16k[1]*p21 + K16k[2]*p22,
                   K25 + K26k[1]*p11 + K26k[2]*p12],
                   variables = [p11, p12, p21, p22])
System of length 4
    4 variables: p11, p12, p21, p22

    -0.902222222222235 - 0.0948148148148107*p21 - 0.0474074074074338*p22
    + 0.669925925925941*p21*p22 + 1.35111111111111*p21^2
    -0.902222222222235 - 0.0948148148148107*p11 - 0.0474074074074338*p12
    + 0.669925925925941*p11*p12 + 1.35111111111111*p11^2
    -0.274814814814818 - 0.0474074074073769*p21 + 3.41786080714587*p22
    -0.274814814814818 - 0.0474074074073769*p11 + 3.41786080714587*p12

julia> sol = solve(F)
Result with 4 solutions
======================
• 4 paths tracked
• 4 non-singular solutions (4 real)
• random_seed: 0x1a96f301
• start_system: :polyhedral


julia> realsol = real_solutions(sol)
4-element Array{Array{Float64,1},1}:
    [0.83152596361697, 0.09193917560505489, -0.8009419096814003,
    0.0692960447358559]
    [-0.8009419096814003, 0.0692960447358559, -0.8009419096814003,
    0.0692960447358559]
    [0.83152596361697, 0.09193917560505489, 0.83152596361697,
    0.09193917560505489]
```

```
    [-0.8009419096814003, 0.0692960447358559, 0.83152596361697,
    0.09193917560505489]
```

I could also have used the interval arithmetic solution method here.

```
julia> using IntervalRootFinding, IntervalArithmetic

julia> using StaticArrays

julia> function f(Ps)
            p11, p12, p21, p22 = Ps

            return SVector{4}([K11 + K12*p21 + K13k[1]*p21^2 +
            K13k[2]*p21*p22 + K14k[2]*p22,
            K21 + K22*p11 + K23k[1]*p11^2 +
            K23k[2]*p11*p12 + K24k[2]*p12,
            K15 + K16k[1]*p21 + K16k[2]*p22,
            K25 + K26k[1]*p11 + K26k[2]*p12])
        end
f (generic function with 1 method)

julia> int01 = interval(0.0, 1.0)
[0, 1]

julia> rts = roots(f, int01 × int01 × int01 × int01, Krawczyk, 1e-12)
1-element Array{Root{IntervalBox{4,Float64}},1}:
    Root([0.831525, 0.831526] × [0.0919391, 0.0919392] ×
     [0.831525, 0.831526] × [0.0919391, 0.0919392], :unique)
```

and again there are no ":unknown" boxes, so I know that this is the only solution.

Above, it can be seen that the homotopy method traced four paths. This is because that method will try to find all solutions even complex ones. The Krawczyk method on the other hand only looks for solutions in the unit box. Either it concludes there are no solutions or it finds the ones that are there. I timed the two solution methods against each other here, and the interval method is consistently five times faster than the homotopy based method.

## G.3 Binary triopoly game

```
julia> K1 = [4.937499999999915, -26.411791666666517, -25.93851091487278,
16.797177581539586, 20.27062499999995, 20.27062499999995, -325.27920833333314]
7-element Array{Float64,1}:
    4.937499999999915
    -26.411791666666517
    -25.93851091487278
    16.797177581539586
    20.27062499999995
    20.27062499999995
    -325.27920833333314

julia> K2 = [4.937499999999915, -26.411791666666517, -20.967499999999944,
31.960855227177035, 20.27062499999995, 20.27062499999995, -344.4071624658179]
7-element Array{Float64,1}:
    4.937499999999915
    -26.411791666666517
    -20.967499999999944
    31.960855227177035
    20.27062499999995
    20.27062499999995
    -344.4071624658179

julia> K3 = [4.937499999999915, -13.937499999999972, -25.938521374040192,
24.593521374040392, 4.690624999999919, 20.27062499999995, -344.5363020833331]
7-element Array{Float64,1}:
    4.937499999999915
    -13.937499999999972
    -25.938521374040192
    24.593521374040392
    4.690624999999919
    20.27062499999995
    -344.5363020833331

julia> @var p1, p2, p3
(p1, p2, p3)
```

```
julia> F = System([K1[1] + K1[2]*p2 + K1[3]*p3 + K1[4]*p2*p3 +
                   K1[5]*p2^2*p3+K1[6]*p2*p3^2+K1[7]*p2^2*p3^2,
                   K2[1] + K2[2]*p1 + K2[3]*p3 + K2[4]*p1*p3 +
                   K2[5]*p1^2*p3+K2[6]*p1*p3^2+K2[7]*p1^2*p3^2,
                   K3[1] + K3[2]*p2 + K3[3]*p1 + K3[4]*p2*p1 +
                   K3[5]*p2^2*p1+K3[6]*p2*p1^2+K3[7]*p2^2*p1^2],
                   variables = [p1, p2, p3])
System of length 3
    3 variables: p1, p2, p3

    4.93749999999991 - 26.4117916666665*p2 - 25.9385109148728*p3 +
    16.7971775815396*p2*p3 + 20.2706249999999*p2*p3^2 +
    20.2706249999999*p2^2*p3 - 325.279208333333*p2^2*p3^2
    4.93749999999991 - 26.4117916666665*p1 - 20.9674999999999*p3 +
    31.960855227177*p1*p3 + 20.2706249999999*p1*p3^2 +
    20.2706249999999*p1^2*p3 - 344.407162465818*p1^2*p3^2
    4.93749999999991 - 25.9385213740402*p1 - 13.9375*p2 +
    24.5935213740404*p2*p1 + 20.2706249999999*p2*p1^2 +
    4.69062499999992*p2^2*p1 - 344.536302083333*p2^2*p1^2

julia> sol = solve(F)
Result with 16 solutions
========================
• 16 paths tracked
• 16 non-singular solutions (8 real)
• random_seed: 0xb313dfc7
• start_system: :polyhedral


julia> realsol = real_solutions(sol)
8-element Array{Array{Float64,1},1}:
    [0.08520268977076134, -5.789199190641236, 0.1475539199235848]
    [-0.5423171911842847, -0.5424548482872174, -0.5824899074436278]
    [0.12664377011175382, 0.14723459253738377, -5.707019344120354]
    [0.21995808549142704, -0.3368657588112441, -0.9894437822983965]
    [-0.6920881397646398, 0.3110406973707669, -0.46525727631097363]
```

```
    [-0.46623443795206726, -0.6255951510600032, 0.30272717141718847]
    [-18.070381007815104, 0.09775486255470776, 0.09728966706072086]
    [0.14054404801015058, 0.11946449646162163, 0.07485594415537336]

julia> using IntervalRootFinding, IntervalArithmetic

julia> using StaticArrays

julia> function f(Ps)
    p1, p2, p3 = Ps
    return SVector{3}([K1[1] + K1[2]*p2 + K1[3]*p3 +
    K1[4]*p2*p3 + K1[5]*p2^2*p3+K1[6]*p2*p3^2+K1[7]*p2^2*p3^2,
                       K2[1] + K2[2]*p1 + K2[3]*p3 +
    K2[4]*p1*p3 + K2[5]*p1^2*p3+K2[6]*p1*p3^2+K2[7]*p1^2*p3^2,
                       K3[1] + K3[2]*p2 + K3[3]*p1 +
    K3[4]*p2*p1 + K3[5]*p2^2*p1+K3[6]*p2*p1^2+K3[7]*p2^2*p1^2])
end
f (generic function with 1 method)

julia> int01 = interval(0.0, 1.0)
[0, 1]

rts = roots(f, int01 × int01 × int01, Krawczyk, 1e-12)
```

Timing these runs HomotopyContinuations.jl comes out faster by a factor of around 2-3x.