Numerical Method for Bertrand Mean Field Games Using Multigrid

by

Peipei Tan

A research paper
presented to the University of Waterloo
in partial fulfillment of the
requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Supervisor: Prof. Justin W.L. Wan

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this report. This is a true copy of the report, including any required final revisions, as accepted by my examiners.

I understand that my report may be made electronically available to the public.

Abstract

We study dynamic Bertrand mean field games (MFG) with exhaustible capacities in which companies compete with each other using the price as the strategic variable and the interaction among the competition is through the average price. We consider both continuum mean field games and finite player games. Dynamic continuum mean field games can be modeled as a system of partial differential equations (PDEs) which consists of a backward Hamilton-Jacobi-Bellman (HJB) equation for the value function and a forward Kolmogorov equation for the density function. This system is coupled through the proportion of remaining companies and the average equilibrium price. Finite N-player games can be approximated using the framework of continuum mean field games. Due to the substantial computational cost of solving the HJB equation, we present an efficient multigrid method with Full Approximation Scheme (FAS) to solve this equation. An iterative algorithm is applied to solve the whole MFG problem. Numerical results illustrate the effects of the competition on the proportion of active companies and the average equilibrium price.

Acknowledgements

I would like to use this opportunity to express my sincere gratitude to my supervisor Prof. Justin W.L. Wan for his invaluable guidance and continuous support during my study and research in the program of Computational Mathematics. He is patient, enthusiastic and a genuinely learned man. Without him, I cannot even image how to complete this thesis. He is one of the most important mentors not only for my master's study but also for my future life.

Besides my advisor, I would also like to thank all instructors for their help with my course study.

My thanks also goes to our lovely administrative coordinators Patty Robinson-Angel and Anthea Dunne who have always given us warm care and useful help during our study here.

And I would like to thank every student in the CM program. I really enjoyed the fun time with you. I will always keep you in mind.

In the end, I give my sincere thanks to my family for supporting me unconditionally throughout my study.

Dedication

This is dedicated to the ones I love.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In economics, business competition is a game among companies who try to increase their profits and achieve higher market share. According to the strategy that companies used to compete with others, there are generally two kinds, Bertrand and Cournot competition.

Cournot competition is named after Antoine Augustin Cournot. In 1838, Cournot first gave this model in his description of a spring water market with two dominating suppliers. In this model, players choose what amount of products to produce as the strategy to win the market share and the market determines the price for each player afterwards.

On the other hand, the idea of Bertrand competition was formulated by Bertrand in a review of Cournot's book in 1883 [5]. It was formalized into a mathematical model by Francis Ysidro Edgeworth in [11]. In this model, companies compete with each other by setting prices and the market determines the demand that each player receives. There was once a debate between Bertrand model and Cournot model in the history [9]. In practice, some cases are better modeled using Cournot while others using Bertrand. It depends on what kind of products are being produced and sold in the market. In our essay, we will concentrate on Bertrand games.

In game theory, if each player knows and sets equilibrium strategies of others, then no player can benefit by changing only its own strategy while others keep their strategies unchanged. This group of strategic choices forms the Nash equilibrium. It has been verified that there exists a unique Nash equilibrium to Bertrand games [18]. In our study, we will analyze Bertrand games in the sense of Nash equilibrium.

Based on the number of players in the competition, it falls into several categories. In historical economics, the market related to resources such as salt, coal and diamonds often

formed a monopoly game in which there is only one player who controls the whole market. However, this kind of competition is usually restricted by government laws in modern times. A much more common game is the duopoly game with two players dominating a market [16]. A good example for the duopoly game is Visa and MasterCard in the market of electronic payment process.

Oligopolies model the market with a small number of players dominating the market. For example, in the bank industry of Canada, there are seven companies (Royal Bank of Canada, Toronto Dominion Bank, Bank of Nova Scotia, Banque de Montreal, Canadian Imperial Bank of Commerce, Desjardins Group and Banque Nationale du Canada) who control the entire banking market. In the literature, there is a large amount of work that has been done for static oligopoly games. It began with Cournot where he took players' strategic interactions into account [8]. In 1983, James Friedman [14] offered a thorough examination of oligopoly theory and he also used some numerical examples to present the idea more explicit. Xavier Vives developed an oligopoly pricing theory based on a modern game-theoretic approach and equilibrium [19]. Beyond the static games, there is also some work developed in the dynamic models. Engwerda [12] built up a linear-quadratic (LQ) model for dynamic problems which holds analytic properties. In [18], the authors studied continuous time Bertrand oligopolies and set up a system of Hamilton-Jacobi-Bellman (HJB) partial differential equations (PDEs) in the sense of linear demand functions. They also illustrated some properties of dynamic games using both asymptotic approximations and numerical analysis.

In the sense of perfect competition, there are a very large number of players in the market and no players can have the dominant power over the market. So each player's impact on the competition can be negligible. Interactions among all sellers act through the average effect. This kind of competition is modelled using mean field games (MFG) proposed by Lasry and Lions [17]. The authors in [7] examined dynamic mean field games. In their work, the continuum game was modeled as a coupled PDE system including a backward HJB equation and a forward Kolmogorov PDE. They gave the asymptotic approximation under several assumptions and also showed some numerical results to further study the equilibrium of the game.

In this essay, we study competitive markets with an infinite number of small players who sell homogeneous and exhaustible products by setting prices as the strategy to win the market share in continuous time. These are modeled using dynamic Bertrand mean field games. Since products are exhaustible, we will use the remaining capacity as the distinct label for each company. In [7], authors solved the HJB equation using the method of lines and the Runge-Kutta method. In our work, we solve the backward HJB equation directly using multigrid methods which is efficient in the sense of the convergent rate.

In Chapter 2, we first present the model for the static version of Bertrand mean field games, and then using the static model, we establish the mathematical model for dynamic mean field games. In Chapters 3 and 4, we give the numerical method to solve the problem including the multigrid method. In the end, some numerical results and conclusions are shown in Chapters 5 and 6.

Chapter 2

Bertrand Mean Field Games

In this chapter, we will present the model for Bertrand mean field games with exhaustible capacities and set up the PDE system for this problem. For better understanding, we will start with static Bertrand games to give some basic concepts and ideas. Then we will study static Bertrand mean field games. In the end, the model for dynamic Bertrand mean field games is given.

2.1 Static Bertrand Games

In this section, we study the static version of Bertrand games with a small number of players. We consider a market with N companies (N is a small number) and each company sets its price as a strategy to compete with other companies. Define $p_i \in \mathbb{R}_+$, $\mathbb{R}_+ = (0, \infty)$ as the price of company $i \in \{1, 2, \ldots, N\}$. The vector p is defined such that the ith element of p is p_i.

In Bertrand games, the market will determine demands for each company's products. For each company $i \in \{1, 2, \ldots, N\}$, we define a demand function $D_i^N(p_1, p_2, \ldots, p_N) : \mathbb{R}_+^N \to \mathbb{R}$ representing the demand it receives from the market. We are going to examine some properties of the demand function in Section 2.1.1 and give a simple linear demand model.

2.1.1 Demand Function and Linear Demand

There are several properties for demand functions.

Property 1 For all $i = 1, 2, \ldots, N$, D_i^N is smooth in all variables, and

$$D_N^N(0, 0, \ldots, 0) > 0, \quad \frac{\partial D_i^N}{\partial p_i} < 0, \quad and \quad \frac{\partial D_i^N}{\partial p_j} > 0 \; for \; i \neq j.$$

This property implies that the demand for one player is decreasing with the player's own price and increasing with the price of its competitors. This is the usual case in the competition with substitute goods which we are considering.

Property 2 For fixed $p_1, p_2, \ldots, p_N$ and all $i, j \in \{1, 2, \ldots, N\}$,

$$D_i^N(p_1, \ldots, p_i, \ldots, p_j, \ldots, p_N) = D_j^N(p_1, \ldots, p_j, \ldots, p_i, \ldots, p_N).$$

Property 3 Fix a vector of prices p. Suppose they are ordered such that $p_1 \leq p_2 \leq \cdots \leq p_N$. Then

$$D_1^N(p_1, \ldots, p_N) \geq D_2^N(p_1, \ldots, p_N) \geq \cdots \geq D_N^N(p_1, \ldots, p_N).$$

This means the order of prices implies the order of demands.

We refer to [18] for these properties and proofs.

For convenience, a linear demand function will be used in our mathematical model and numerical analysis for Bertrand games. Since under our assumptions, the demand for each company only depends on its own price and the sum of its competitors' prices, the linear function for $D_i^N(p_1, p_2, \ldots, p_N)$ can be defined as

$$D_i^N(p_1, p_2, \ldots, p_N) \triangleq A - Bp_i + C \sum_{j \neq i} p_j, \quad i = 1, 2, \ldots, N. \tag{2.1}$$

In this definition, the positive parameters A, B and C satisfy the condition that $B > (N-1) \cdot C$. We can easily see that the demand function in (2.1) meets Properties 1 and 2 and more validations for this linear form can be found in Section 2.2 of [18].

2.1.2 Profit Function and Nash Equilibrium

In Bertrand competition, each company $i \in \{1, 2, \ldots, N\}$ comes with a marginal cost s_i and a vector s is defined as $s = (s_1, s_2, \ldots, s_N)$. With the price p_i and the demand function $D_i^N(p)$, the profit function $\prod_i(p_1, p_2, \ldots, p_N, s_i)$ for company i can be given by

$$\prod_i (p_1, p_2, \ldots, p_N, s_i) \triangleq D_i^N(p) \cdot (p_i - s_i). \tag{2.2}$$

In the competition, each player wants to choose a price so that its profit can be maximized. In order to do so, it also needs to take other companies' action into account.

Define a vector $p^* = (p_1^*, p_2^*, \ldots, p_N^*)$ as a Nash equilibrium of the Bertrand games. Then for all $i \in \{1, 2, \ldots, N\}$ we have

$$p_i^* = \arg\max_{p \geq s_i} \prod_i (p_1^*, p_2^*, \ldots, p_{i-1}^*, p, p_{i+1}^*, \ldots, p_N^*, s_i). \tag{2.3}$$

Equation (2.3) shows that the price of the Nash equilibrium is the best decision player i can make while considering all others' decisions. Details for the existence and construction of Nash equilibrium for Bertrand games are illustrated in [18]. We will continue our analysis for Bertrand games in the sense of Nash equilibrium in the following essay.

2.2 Static Bertrand Mean Field Games

From Section 2.1, we have got some basic concepts for Bertrand games. In this section, these concepts will be used to study the static model for Bertrand mean field games which is necessary in the establishment of the PDE system for dynamic games.

First we will analyze a discrete N-player game where there are N players in the market and N is a very large number. Later on we will study the continuum case when $N \to \infty$.

2.2.1 Static N-player Games

For convenience, we assume all N players receive positive demands. Notations for p_i, p, $D_i^N(p)$, and s_i are defined as in Section 2.1. Since we are now considering a market with a very large number of small interacting companies, no company can affect the market individually. The interaction among the competition is through the average price. Similar to (2.1) a linear demand function for N-player games can be defined as

$$D_i^{(N)}(p) = a_N - b_N p_i + c_N \bar{p}_i, \quad \bar{p}_i = \frac{1}{N-1} \sum_{j \neq i}^{N} p_j, \tag{2.4}$$

where a_N, b_N and c_N are positive parameters such that $a_N + c_N = b_N$. From this definition, we can see the demand that one company receives decreases with its own price but increases with the average price of other companies. This means the interaction among the competition acts through the average effect in the market.

Then the profit function for player i can be defined as

$$\prod_i(p_1, p_2, \ldots, p_N, s_i) \triangleq D_i^N(p)(p_i - s_i) = (a_N - b_N p_i + c_N \bar{p}_i)(p_i - s_i). \tag{2.5}$$

Each player wants to maximize its profits during the competition. This behavior can be expressed as an optimization problem which is

$$\max_{p_i}(a_N - b_N p_i + c_N \bar{p}_i)(p_i - s_i). \tag{2.6}$$

The optimal price for player i can be obtained using the first order condition for this optimization problem. It is given as

$$p_i^* = \frac{1}{2b_N}(a_N + b_N s_i + c_N \bar{p}_i), \quad i = 1, 2, \ldots, N. \tag{2.7}$$

If $\bar{p}_i$ in (2.7) is the average of all the other companies' optimal prices, a Nash equilibrium $(p_1^*, p_2^*, \ldots, p_N^*)$ for static N-player games can be achieved and we have

$$\bar{p}_i = \frac{1}{N-1}\sum_{j \neq i} p_j^*.$$

2.2.2 Static Continuum Mean Field Games

As $N \to \infty$, static continuum mean field game is formed. In this case, players are labeled with their capacities $x > 0$ as different sizes. The marginal cost is defined as $s(x)$. Since there are an infinite number of players in the market, a density function $M(x)$ is needed to model the distribution of companies with different capacities.

We define a variable $\eta \in [0, 1]$ to represent the proportion of players that remain in the market. The player with capacity x sets its price to be p as a strategy in the competition. Since the number of companies in the market is so large, one individual's price won't affect the average price of the whole market any more. So we can define $\bar{p}$ as the average price of all players in the market. The demand function for the static continuum MFG can be given as

$$D^{(\eta)}(p, \bar{p}) = a(\eta) - b(\eta)p + c(\eta)\bar{p}. \tag{2.8}$$

An explicit solution for $a(\eta)$, $b(\eta)$ and $c(\eta)$ is

$$a(\eta) = \frac{1}{(1 + \epsilon\eta)}, \quad b(\eta) = 1, \quad c(\eta) = \frac{\epsilon\eta}{(1 + \epsilon\eta)}, \tag{2.9}$$

where ϵ measures the degree of the interaction among the competition in the market. When $\epsilon = 0$, there is not any competition in the market and companies are selling independent products.

The profit function for the player with capacity x is given as

$$\prod(p, \bar{p}) \triangleq D^{(\eta)}(p, \bar{p})(p - s(x)) = \big(a(\eta) - b(\eta)p + c(\eta)\bar{p}\big)\big(p - s(x)\big). \tag{2.10}$$

The optimization problem for this player is

$$\max_{p} \big(a(\eta) - p + c(\eta)\bar{p}\big)\big(p - s(x)\big). \tag{2.11}$$

The first order condition gives the optimal price as

$$p^*(x) = \frac{1}{2}\big(s(x) + a(\eta) + c(\eta)\bar{p}\big). \tag{2.12}$$

In the sense of Nash equilibrium the average price $\bar{p}$ can be given as

$$\bar{p} = \int_0^\infty p^*(x)M(x)dx. \tag{2.13}$$

Multiplying (2.12) by $M(x)$ and then integrating over x gives

$$\bar{p} = \frac{1}{2 - c(\eta)}\big(\bar{s} + a(\eta)\big), \quad where \quad \bar{s} = \int_0^\infty s(x)M(x)dx. \tag{2.14}$$

2.3 Dynamic Bertrand Games

Before moving to dynamic Bertrand mean field games, we first give a study of dynamic Bertrand games with a small number of players [18] to get some essential ideas about differential games.

Consider a market with N (N is a small number) players who set their strategies dynamically over time. At time $t = 0$, player $i \in \{1, 2, \ldots, N\}$ has a capacity of production defined as $x_i(0)$. As time evolves, the capacity of player i will be consumed gradually. Function $x_i(t)$ is denoted as the remaining capacity at time t for player i. When $x_i(t) = 0$, the company exhausts its capacity and drops out of the market. Let $\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_N(t))$. The dynamic price strategy that player i chooses is defined as $p_i(\mathbf{x}(t))$. Given all prices, the

demand determined by the market for player i is denoted by $D_i^N(p_1, p_2, \ldots, p_N)$. Since actual demands are also subject to random fluctuations, the demand can be modeled simply as

$$d_i(t) = D_i^N(p_1, p_2, \ldots, p_N) - \sigma_i \dot{\epsilon}_i(t) \tag{2.15}$$

where $\{\dot{\epsilon}_i(t)\}_{i=1,2,\ldots,N}$ are correlated Gaussian white noises. Then the capacity of player i follows the dynamics given by

$$dx_i(t) = -D_i^N(p_1(\mathbf{x}(t)), p_2(\mathbf{x}(t)), \ldots, p_N(\mathbf{x}(t)))dt + \sigma_i dW_i(t), \tag{2.16}$$

if $x_i > 0,\ i = 1, 2, \ldots, N$, and $\{W_i(t)\}_{i=1,2,\ldots,N}$ are Brownian motions. If $x_i(t) = 0$, then $x_i(s) = 0$ for all $s \geq t$. This means a company can not be revived by random fluctuations when it exhausts its capacity [2].

Now we use a duopoly game with linear demands as an example to show some explicit ideas about dynamic Bertrand games. In the duopoly game, where $N = 2$, assuming linear demands $D_i(p_1, p_2)$, the expected lifetime profit discounted at the constant rate r for player $i \in \{1, 2\}$ is defined as

$$\mathbb{E}\Big\{ \int_0^\infty e^{-rt} p_i(\mathbf{x}(t)) D_i\big(p_1(\mathbf{x}(t)), p_2(\mathbf{x}(t))\big) \mathbb{1}_{\{x_i(t)>0\}} dt \Big\}, \tag{2.17}$$

where the indicator function $\mathbb{1}_{\{x_i(t)>0\}} : x_i(t) \to \{0, 1\}$ is defined as:

$$\mathbb{1}_{\{x_i(t)\}>0} := \begin{cases} 1 & \text{if } x_i(t) > 0, \\ 0 & \text{if } x_i(t) \leq 0. \end{cases}$$

Since both players want to maximize their profits, value functions of these two companies can be defined as a coupled optimization problem which is given by

$$V_i(x_1, x_2) = \sup_{p_i \geq 0} \mathbb{E}\Big\{ \int_0^\infty e^{-rt} p_i(\mathbf{x}(t)) D_i\big(p_1(\mathbf{x}(t)), p_2(\mathbf{x}(t))\big) \mathbb{1}_{\{x_i(t)>0\}} dt \Big\},\ i = 1, 2. \tag{2.18}$$

By a usual dynamic programming argument [13], with sufficient regularity these value functions satisfy the PDE system

$$\mathcal{L}V_i + \sup_{p_i \geq 0} \Big\{ -D_1(p_1, p_2)\frac{\partial V_i}{\partial x_1} - D_2(p_1, p_2)\frac{\partial V_i}{\partial x_2} + p_i D_i(p_1, p_2) \Big\} - rV_i = 0,\ i = 1, 2, \tag{2.19}$$

where

$$\mathcal{L} = \frac{1}{2}\sigma_1^2 \frac{\partial^2}{\partial x_1^2} + \rho\sigma_1\sigma_2 \frac{\partial^2}{\partial x_1 \partial x_2} + \frac{1}{2}\sigma_2^2 \frac{\partial^2}{\partial x_2^2},$$

and ρ is the correlation coefficient of the Brownian motions: $\mathbb{E}\{dW_1 dW_2\} = \rho dt$.

As we can see, in order to solve these differential games we need to solve a system of PDEs and this can be very difficult. Related work has been done in [3, 4] where Bensoussan and Frehse presented a useful application of the regularity theory for non-linear PDE systems to solve stochastic differential games with N players.

2.4 Dynamic Bertrand Mean Field Games

In this section, we are going to analyse Dynamic Bertrand mean field games [7] in which an infinite number of companies with different sizes sell substitute and exhaustible products over time. We firstly give the framework for dynamic continuum mean field games. Then we use that framework to approximate dynamic N-player games.

2.4.1 Dynamic Continuum Mean Field Games

Consider a market with innumerable companies. These companies can be distinguished between bigger and smaller players based on the initial capacities of the production. At $t = 0$, $M(x)$ is defined as the density function of players with capacity $x > 0$. In continuous time games, the capacity of one player will be consumed over time. Let $X(t)$ denote the remaining capacity at time t. Function $m(t, x)$ is used to denote the density of companies with positive capacity x at time t. The proportion of remaining active players at time t can be defined as:

$$\eta(t) = \int_{\mathbb{R}_+} m(t, x)dx. \tag{2.20}$$

Players in dynamic mean field games set prices as the strategy over time and this price strategy can be defined as $p(t, x)$ at time t. From Section 2.2.2, demands that players receive from the market are given as

$$q(t, x) = D^{\eta(t)}\big(p(t, x), \bar{p}(t)\big) = a\big(\eta(t)\big) - p(t, x) + c\big(\eta(t)\big)\bar{p}(t), \tag{2.21}$$

where functions $a\big(\eta(t)\big)$ and $c\big(\eta(t)\big)$ are defined in (2.9). The average price $\bar{p}$ at time t is given by:

$$\bar{p}(t) = \frac{1}{\eta(t)} \int_{\mathbb{R}_+} p(t, x)m(t, x)dx. \tag{2.22}$$

As developed in Section 2.3, the actual remaining capacity $X(t)$ for the stochastic differential games is modeled as:

$$dX(t) = -q\big(t, X(t)\big)dt + \sigma \mathbb{1}_{\{X(t)>0\}} dW_t, \tag{2.23}$$

where W is the Brownian motion and σ is a non-negative constant.

Now we are ready to give the value function. A company with remaining capacity $x > 0$ uses the price as the strategy to achieve optimal lifetime profits. The profits are discounted at a constant rate $r > 0$ over Markov controls $p_t = p\big(t, X(t)\big)$. In this sense, the value function of the company is given by:

$$u(t,x) = \sup_p \mathbb{E}\bigg\{ \int_t^T e^{-r(s-t)} p_s q_s ds | X(t) = x \bigg\}, \quad x > 0. \tag{2.24}$$

The associated HJB equation is

$$u_t + \frac{1}{2}\sigma^2 u_{xx} - ru + \max_{p\geq 0} \bigg[\Big(a\big(\eta(t)\big) - p + c\big(\eta(t)\big)\bar{p}(t) \Big) \Big(p - u_x \Big) \bigg] = 0. \tag{2.25}$$

When a player exhausts its capacity, it can no longer earns profits. So we have $u(t,0) = 0$. At the terminal time T, all companies' reserves will be exhausted, so $u(T,x) = 0$. The shadow cost says that $u' \to 0$ as $x \to 0$.

Using the first order condition for (2.25) we will get the optimal price which is given by

$$p^*(t,x) = \frac{1}{2}\Big(a\big(\eta(t)\big) + u_x(t.x) + c\big(\eta(t)\big)\bar{p}(t) \Big).$$

The optimal price can be used to compute the average equilibrium price in (2.22) and we can get

$$\bar{p}(t) = \frac{1}{2 - c(\eta(t))} \Big(a\big(\eta(t)\big) + \frac{1}{\eta(t)} \int_{\mathbb{R}_+} u_x(t,x) m(t,x) dx \Big).$$

Then the HJB equation can be written as

$$u_t + \frac{1}{2}\sigma^2 u_{xx} - ru + \frac{1}{4}\Big(a\big(\eta(t)\big) - u_x + c\big(\eta(t)\big)\bar{p}(t) \Big)^2 = 0. \tag{2.26}$$

From (2.21) we can obtain the following optimal equilibrium demand function as

$$q^*(t,x) = \frac{1}{2}\bigg(a\big(\eta(t)\big) - u_x(t,x) + c\big(\eta(t)\big)\bar{p}(t) \bigg). \tag{2.27}$$

Meanwhile, the density function $m(t,x)$ of the distribution of the remaining capacity $X(t)$ is the solution to the following forward Kolmogorov equation

$$m_t - \frac{1}{2}\sigma^2 m_{xx} + \Big(-\frac{1}{2}\Big(a\big(\eta(t)\big) - u_x + c\big(\eta(t)\big)\bar{p}(t)\Big)m\Big)_x = 0, \tag{2.28}$$

with initial condition $m(0,x) = M(x)$.

Depending on the value of σ, the problem can be distinguished as two cases: stochastic and deterministic.

1. When $\sigma = 0$, there is no random fluctuations in the market and the demand function is deterministic. There is no boundary condition at $x = 0$.

2. When $\sigma > 0$, once a player reaches zero it cannot be revived back to the market due to the existence of Brownian motion. So we have the boundary condition as $m(t,0) = 0$.

The system formed by (2.25) and (2.28) is an instance model of dynamic Bertrand mean field games. This system is coupled through the proportion function $\eta(t)$ and the average price function $\bar{p}(t)$. Although some regularity results exist [17] for this problem, fully solving it is still an ongoing challenge.

2.4.2 Dynamic N-player Games

In practice, there are usually a finite number of companies in the market competing with each other in continuous time. This is N-player games where N is a very large number. The framework of continuum mean field games can be used to study these N-player games. The idea is to approximate the initial density function $M(x)$ using the sum of Dirac-delta functions:

$$M(x) = \frac{1}{N}\sum_{i=1}^{N} \delta(x - X_i^0), \quad i \in \{1, 2, \ldots, N\}, \tag{2.29}$$

where δ is the Dirac-delta function and X_i^0 is the initial capacity of company i.

We consider the deterministic case for N-player games. As defined in (2.23), when $\sigma = 0$, the consumption of the capacity is only subject to the demand which is $dX(t) = -q(t, X(t))dt$. So we have the following ODE system for N players

$$\frac{d}{dt}X_i(t) = -q^*(t, X_i(t)),\ X_i(0) = X_i^0,\ i \in \{1, 2, \ldots, N\}, \tag{2.30}$$

where $q^*(t,x)$ is the optimal equilibrium demand function defined in (2.27).

In this deterministic setting, there is a hitting time τ^i at which player i exhausts its capacity which means $X_i(\tau^i) = 0$. In this sense the proportion of remaining companies $\eta(t)$ is a pure jump function. Assuming we have obtained the hitting time for all companies and sorted them in the order such that $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_N$. For $\tau_{k-1} < t \leq \tau_k$, the proportion function can be defined as

$$\eta(t) = \frac{N-k+1}{N}. \tag{2.31}$$

And the average price function $\bar{p}(t)$ is

$$\bar{p}(t) = \frac{1}{2 - c\big(\eta(t)\big)}\Big(a\big(\eta(t)\big) + \frac{1}{\eta(t)}\frac{1}{N}\sum_{i=k}^{N} u_x\big(t, X_i(t)\big)\Big). \tag{2.32}$$

In the next chapter, we will discuss our numerical method to solve this problem.

Chapter 3

Numerical Method for Mean Field Games

We develop our numerical method for solving dynamic Bertrand mean field games inspired by the solution strategy in [7]. In their work, an iterative method was used to solve the PDE system. While in our work, overall we also use an iterative algorithm to solve the problem, however, with different schemes to solve each individual PDE in the system, in particular, the backward HJB equation. First we will discuss the continuum case, and later we apply the continuum MFG framework to approximate N-player games.

3.1 Numerical Method for Continuum Mean Field Games

The continuum MFG problem consists of several coupled equations. As discussed in [7], the density function $m(t, x)$ turns out to be singular when $x \to 0$. This will raise difficulties for the numerical solver. Hence the tail distribution function is introduced to achieve better numerical treatment and it is defined as

$$\bar{\eta}(t, x) = \int_x^\infty m(t, y)dy. \tag{3.1}$$

From the definition of the tail distribution function, we have $m(t, x) = -\bar{\eta}_x(t, x)$ and $\eta(t) = \int_0^\infty m(t, y)dy = \bar{\eta}(t, 0)$.

Substituting (3.1) into the forward Kolmogorov equation in (2.28), we obtain the fol-

lowing equation

$$\begin{aligned}
&\bar{\eta}_t(t,x) - \frac{1}{2}\sigma^2\bar{\eta}_{xx}(t,x) - \frac{1}{2}\Big(a\big(\eta(t)\big) - u_x(t,x) + c\big(\eta(t)\big)\bar{p}(t)\Big)\bar{\eta}_x(t,x) = 0,\\
&\bar{\eta}(0,x) = \int_x^\infty M(y)dy.
\end{aligned} \tag{3.2}$$

When $\sigma > 0$, we choose boundary conditions for (3.2) such that

$$\lim_{x\to\infty}\bar{\eta}(t,x) = 0, \quad \lim_{x\to 0}\bar{\eta}_x(t,x) = 0.$$

The reason is that once a company exhausts its capacity it is out of the competition.

Now our model equations become

$$\begin{aligned}
&(a) \quad u_t + \frac{1}{2}\sigma^2 u_{xx} - ru + \max_{p\geq 0}\left\{\Big(a\big(\eta(t)\big) - p + c\big(\eta(t)\big)\bar{p}(t)\Big)\Big(p - u_x\Big)\right\} = 0,\\
&(b) \quad \bar{\eta}_t(t,x) - \frac{1}{2}\sigma^2\bar{\eta}_{xx}(t,x) - \frac{1}{2}\Big(a\big(\eta(t)\big) - u_x(t,x) + c\big(\eta(t)\big)\bar{p}(t)\Big)\bar{\eta}_x(t,x) = 0,\\
&(c) \quad \eta(t) = \bar{\eta}(t,0),\\
&(d) \quad m(t,x) = -\bar{\eta}_x(t,x),\\
&(e) \quad \bar{p}(t) = \frac{1}{\eta(t)}\int_{\mathbb{R}_+} p^*(t,x)m(t,x)dx,
\end{aligned} \tag{3.3}$$

where functions $a\big(\eta(t)\big), c\big(\eta(t)\big)$ are as defined in (2.9) from Chapter 2.

It is very difficult to have analytic solutions for these equations. In [7], authors gave analytic solutions for this system in the case that the demand function is deterministic ($\sigma = 0$) and no competition exists in the market ($\epsilon = 0$). However, our ultimate goal is to evaluate effects of the market competition on mean field games. So a numerical method will generally be necessary to solve this problem. We shall start the description of our numerical method with the discretization for all model equations.

3.1.1 Discretization

In the discretization for our model equations, we will use a uniform grid $x_0, x_1, \ldots, x_S$ with $x_i = x_0 + i\cdot\Delta x$, where x_S is the right boundary of our interval, Δx is the grid size and x_i is the ith grid point, $i \in \{0, 1, 2, \ldots, S\}$. For the time stepping, a time discretization $0 = t^0 < t^1 < t^2 < \cdots < t^L = T$ is defined with time step Δt, where T is the terminal time when all the companies exhaust their reserves, and $t^{n+1} = t^n + \Delta t$.

Discretization for the HJB equation

As observed, the HJB equation in (3.3)-(a) is a backward equation, and we have the terminal condition $u(T,x)=0$ instead of the initial condition. Since terms $\frac{1}{2}\sigma^2 u_{xx}$ and ru do not depend on the control, the HJB equation can be rewritten as

$$-u_t = \max_{p\geq 0}\left\{\frac{1}{2}\sigma^2 u_{xx} - ru + \Big(a\big(\eta(t)\big) - p + c\big(\eta(t)\big)\bar{p}(t)\Big)\Big(p - u_x\Big)\right\}. \tag{3.4}$$

Let u_i^n be the discrete approximation to $u(t^n,x_i)$, η^n be the approximate solution to $\eta(t^n)$ and $\bar{p}^n$ be the approximate solution to $\bar{p}(t^n)$. Let $u^n=[u_o^n,u_1^n,\ldots,u_S^n]$. Using finite difference methods and implicit time stepping we can get the discretization for (3.3)-(a) which is given by

$$\begin{aligned}-\frac{u_i^{n+1}-u_i^n}{\Delta t} = \max_{p\geq 0}\Bigg\{&\frac{1}{2}\sigma^2\frac{u_{i+1}^n-2u_i^n+u_{i-1}^n}{\Delta x^2} - ru_i^n \\ &+ \big(a(\eta^n) - p + c(\eta^n)\bar{p}^n\big)\big(p - \frac{u_{i+1}^n-u_i^n}{\Delta x}\big)\Bigg\}.\end{aligned} \tag{3.5}$$

Let $A_i^n = \big(a(\eta_i^n) - p + c(\eta_i^n)\bar{p}_i^n\big)$. Rearranging the discretized equation in (3.5) gives

$$\frac{u_i^n-u_i^{n+1}}{\Delta t} = \max_{p\geq 0}\left\{\frac{\sigma^2}{2\Delta x^2}u_{i-1}^n + (\frac{\sigma^2}{2\Delta x^2} - \frac{A_i^n}{\Delta x})u_{i+1}^n - (\frac{\sigma^2}{\Delta x^2} + r + \frac{A_i^n}{\Delta x})u_i^n + A_i^n p\right\}. \tag{3.6}$$

Let

$$\alpha_i^n = \frac{\sigma^2}{2\Delta x^2}, \quad \beta_i^n = \frac{\sigma^2}{2\Delta x^2} - \frac{A_i^n}{\Delta x}, \quad \gamma_i^n = \frac{\sigma^2}{\Delta x^2} + r + \frac{A_i^n}{\Delta x}.$$

Then we can write our discretization as

$$\frac{u_i^n-u_i^{n+1}}{\Delta t} = \max_{p\geq 0}\left\{\alpha_i^n u_{i-1}^n + \beta_i^n u_{i+1}^n - \gamma_i^n u_i^n + A_i^n p\right\}. \tag{3.7}$$

Note that u_i^{n+1} is known while u_i^n is what we should compute from this equation.

Discretization for $\bar{\eta}(t,x)$

Let $\bar{\eta}_i^n$ be the approximate solution to $\bar{\eta}(t^n,x_i)$. Using finite difference methods and implicit time stepping we get the discretized equation for $\bar{\eta}_i^n$ which is

$$\frac{\bar{\eta}_i^{n+1}-\bar{\eta}_i^n}{\Delta t} = \theta_i^{n+1}\bar{\eta}_{i-1}^{n+1} + \kappa_i^{n+1}\bar{\eta}_{i+1}^{n+1} - (\theta_i^{n+1}+\kappa_i^{n+1})\bar{\eta}_i^{n+1}, \tag{3.8}$$

where

$$\theta_i^{n+1} = \frac{\sigma^2}{2\Delta x^2}, \quad \kappa_i^{n+1} = \frac{\sigma^2}{2\Delta x^2} + \frac{1}{2\Delta x}\big(a(\eta^{n+1}) - u_i'^{n+1} + c(\eta^{n+1})\bar{p}^{n+1}\big),$$

and $u_i'^{n+1}$ is the approximate solution to $u_x(t^{n+1}, x_i)$.

Discretization for $\eta(t)$

Let η^n be the approximate solution to $\eta(t^n)$. Since $\eta(t^n) = \bar{\eta}(t^n, 0)$, we can simply compute

$$\eta^n = \bar{\eta}_0^n. \tag{3.9}$$

Discretization for the density function $m(t, x)$

Let m_i^n be the approximate solution to $m(t^n, x_i)$. We will use a centered difference method to approximate the derivative of $\bar{\eta}(t, x)$ for the interior grid points which is

$$m_i^n = -\frac{\bar{\eta}_{i+1}^n - \bar{\eta}_{i-1}^n}{2\Delta x}, \quad i = 1, 2, \ldots, S-1. \tag{3.10}$$

On boundaries, we use forward and backward difference methods instead:

$$\begin{aligned} m_0^n &= -\frac{\bar{\eta}_1^n - \bar{\eta}_0^n}{\Delta x}, \\ m_S^n &= -\frac{\bar{\eta}_S^n - \bar{\eta}_{S-1}^n}{\Delta x}. \end{aligned} \tag{3.11}$$

Discretization for the average price function $\bar{p}(t)$

Let p_i^{*n} be the approximate solution to $p^*(t^n, x_i)$ which is the optimal price for the grid point x_i at time t^n. Define a vector p^{*n} such that $p^{*n} = [p_i^{*n}, p_2^{*n}, \ldots, p_S^{*n}]$. Let $m^n = [m_1^n, m_2^n, \ldots, m_S^n]$. The discretization for $\bar{p}$ is

$$\bar{p}^n = \frac{1}{\eta^n} \int_{\mathbb{R}_+} p^{*n} \cdot m^n dx. \tag{3.12}$$

Now we have all discretized equations for our problem system. For clarity, we use (u, $\bar{\eta}$, η, m, $\bar{p}$) to represent discrete approximations to ($u(t, x)$, $\bar{\eta}(t, x)$, $\eta(t)$, $m(t, x)$, $\bar{p}(t)$). Before we talk about numerical methods for solving all these discretized equations, we need to give the whole framework of the numerical method to solve the continuum MFG problem.

3.1.2 Framework of Numerical Method for Continuum MFG

As we mentioned before, these discretized equations are coupled with each other through η and $\bar{p}$. We will apply an iterative method to solve this problem. The natural idea is to start with some initial guesses for η and $\bar{p}$, and then use these initial guesses to solve these discretized equation for u and $\bar{\eta}$. After that we can update solutions for η and $\bar{p}$ using new values for u and $\bar{\eta}$. We iterate these steps until differences between solutions to all four unknowns $(u, \bar{\eta}, \eta, \bar{p})$ from the kth and $(k+1)$th iterations are close enough. In this way, we can obtain an approximate solution to the continuum MFG problem.

Note that the HJB equation for u is backward while the equation for $\bar{\eta}$ is forward. We cannot solve these two equations at the same time. Hence firstly we need to solve the backward HJB equation for all time steps and then use the numerical result obtained for u to solve the forward equation for $\bar{\eta}$ for all time steps. In each time step, we solve equations for both u and $\bar{\eta}$ using numerical solvers until convergence is achieved.

In the next section, we will give numerical methods for solving the individual discretized equation derived in Section (3.1.1).

3.1.3 Numerical Method for Each Equation

Numerical method for u

As we can see the discretized equation (3.7) is highly non-linear. In [15], authors used a multigrid method with Full Approximation Scheme(FAS) to solve HJB equations and results showed that this method is efficient in the sense that the number of iterations used to converge was insensitive to the grid size. So we will apply this multigrid method to solve our HJB equation. For better understanding, it is useful to give some introductory theory about multigrid methods before we describe our multigrid solver for the HJB equation. As a result, we will use the next chapter to discuss the multigrid algorithm with FAS for solving the HJB equation.

Numerical method for $\bar{\eta}$

Given the linear system of component form in (3.8) for $\bar{\eta}$, one method for solving these equations is a relaxation scheme. Rearranging equation (3.8) and noting that θ_i^{n+1}, κ_i^{n+1}

are all non-negative, we obtain,

$$\bar{\eta}_i^{n+1} = \frac{\bar{\eta}_i^n + \Delta t(\theta_i^{n+1}\bar{\eta}_{i-1}^{n+1} + \kappa_i^{n+1}\bar{\eta}_{i+1}^{n+1})}{1 + \Delta t(\theta_i^{n+1} + \kappa_i^{n+1})}. \tag{3.13}$$

Let $\hat{\bar{\eta}}^{k+1}$ be the $(k+1)$th estimate for $\bar{\eta}^{n+1}$. Equation (3.14) can be used as a basis for the relaxation scheme

$$\hat{\bar{\eta}}_i^{k+1} = \frac{\bar{\eta}_i^n + \Delta t(\theta_i^{n+1}\hat{\bar{\eta}}_{i-1}^{k} + \kappa_i^{n+1}\hat{\bar{\eta}}_{i+1}^{k})}{1 + \Delta t(\theta_i^{n+1} + \kappa_i^{n+1})}. \tag{3.14}$$

We apply this scheme iteratively to solve the equation for $\bar{\eta}$.

Numerical method for η

The proportion function η can be simply updated using the numerical solution for $\bar{\eta}$ and the formula is as defined in (3.9).

Numerical method for the density function m

The numerical solution for m can also be updated using the numerical result for $\bar{\eta}$. This operation is performed in the way defined in our discretized equations in (3.10) and (3.11).

Numerical method for the average price function $\bar{p}$

For computational convenience, we use the Matlab built-in function "trapz" to compute the numerical integration for I^n and update the solution to $\bar{p}$ using the formula in (3.13).

3.1.4 Summary of Numerical Method for Continuum MFG

The whole iterative scheme is described in Algorithm 1. In this algorithm, the input is the initial condition $\bar{\eta}^0$ for $\bar{\eta}$, the terminal condition u^L for u, and initial guesses $\hat{\eta}^0$, $\hat{\bar{p}}^0$ for functions η and $\bar{p}$. The output is an numerical solution $(\hat{u}, \hat{\eta}, \hat{\bar{p}})$ to $(u, \eta, \bar{p})$ of our MFG problem. The tolerance for the stopping criterion is usually $tol = 10^{-6}$.

Algorithm 1 $(\hat{u}, \hat{\eta}, \hat{\bar{p}}) = MFGsolver(u^L, \bar{\eta}^0, \hat{\eta}^0, \hat{\bar{p}}^0)$

Set: $\hat{\eta}^k = \hat{\eta}^0$, $\hat{\bar{p}}^k = \hat{\bar{p}}^0$.

while $\{(\hat{u}^{k+1} - \hat{u}^k > tol)$ or $(\hat{\eta}^{k+1} - \hat{\eta}^k > tol)$ or $(\hat{\bar{p}}^{k+1} - \hat{\bar{p}}^k > tol)\}$ **do**

Step 1 Let $\hat{u}^k = \hat{u}^{k+1}$, $\hat{\eta}^k = \hat{\eta}^{k+1}$ and $\hat{\bar{p}}^k = \hat{\bar{p}}^{k+1}$.

Step 2 Using $(\hat{\eta}^k,\ \hat{\bar{p}}^k)$ to solve the discretized equation in (3.7) for u for all time steps and obtain the approximate solution $\hat{u}^{k+1}$.

Step 3 Using $(\hat{u}^{k+1},\ \hat{\eta}^k,\ \hat{\bar{p}}^k)$ to solve the discretized equation in (3.8) for $\bar{\eta}$ for all time steps and obtain the approximate solution $\hat{\bar{\eta}}^{k+1}$.

Step 4 Update the numerical solution to the proportion function η using the method in (3.9) and obtain $\hat{\eta}^{k+1}$.

Step 5 Compute the approximate solution to the density function m using formulas in (3.10) and (3.11) and obtain $\hat{m}^{k+1}$.

Step 6 Update the solution to the average price function $\bar{p}$ using formula (e) in (3.3) and obtain $\hat{\bar{p}}^{k+1}$.

end while

Return: $(\hat{u}^{k+1}, \hat{\eta}^{k+1}, \hat{\bar{p}}^{k+1})$.

3.2 Numerical Method for N-player Games

The model equations for the deterministic N-player games are

$$
\begin{aligned}
&(a) \quad u_t + \frac{1}{2}\sigma^2 u_{xx} - ru + \max_{p \geq 0} \left\{ \Big(a\big(\eta(t)\big) - p + c\big(\eta(t)\big)\bar{p}(t)\Big)\Big(p - u_x\Big) \right\} = 0, \\
&(b) \quad \frac{d}{dt}X_i(t) = -q^*\big(t, X_i(t)\big), X_i(0) = X_i^0, \quad i \in \{1, 2, \ldots, N\}, \\
&(c) \quad \eta(t) = \frac{N - k + 1}{N}, \quad for \quad \tau_{k-1} < t \leq \tau_k, \\
&(d) \quad \bar{p}(t) = \frac{1}{2 - c\big(\eta(t)\big)}\Big(a\big(\eta(t)\big) + \frac{1}{\eta(t)}\frac{1}{N}\sum_{i=k}^{N} u_x\big(t, X_i(t)\big)\Big), \quad for \quad \tau_{k-1} < t \leq \tau_k,
\end{aligned}
\tag{3.15}
$$

where equation (b) is the ODE for the capacity function and τ_k is the hitting time as denoted in Section 2.4.2. Since we consider the deterministic case for N-player games, there is no Brownian motions in ODEs and the remaining capacity only depends on demands for company's products.

We will use the framework for the continuum MFG problem to study N-player games.

3.2.1 Discretization

The same uniform grid and time discretization are defined with those in Section 3.1.1.

Discretization for the HJB equation

Since (3.15) (a) is the same HJB equation as that in the continuum MFG problem, we can just use the discretization from that part.

Discretization for ODEs

Let X_i^n be the approximate solution to $X_i(t^n)$. Performing time stepping to ODEs we get the discretized equation for company i

$$\frac{X_i^{n+1} - X_i^n}{\Delta t} = -q^*(t^n, X_i^n). \tag{3.16}$$

Discretization for the proportion function $\eta(t)$

Let η^n denote the approximation to $\eta(t^n)$. Then for $\tau_{k-1} < t^n \leq \tau_k$,,

$$\eta^n = \frac{N-k+1}{N}. \tag{3.17}$$

Discretization for the average price function $\bar{p}(t)$

Let $\bar{p}^n$ be the approximate solution to $\bar{p}(t^n)$ and $u'^n(x)$ be the approximate derivative of u^n. Then the discretized equation for (d) will be

$$\bar{p}^n = \frac{1}{2 - c(\eta^n)} \Big(a(\eta^n) + \frac{1}{\eta^n} \frac{1}{N} \sum_{i=k}^{N} u'^n(X_i^n) \Big). \tag{3.18}$$

3.2.2 Framework of Numerical Method for N-player Games

The numerical scheme for N-player games also relies on an iterative algorithm starting with some initial guesses $\hat{\eta}^0, \hat{\bar{p}}^0$ for $\eta, \bar{p}$. Using $\hat{\eta}^0, \hat{\bar{p}}^0$ we solve the HJB equation. Next we solve ODEs for all time steps. After that the solution to η and $\bar{p}$ can be updated. These steps are iterated until the stopping criterion is satisfied.

3.2.3 Numerical Method for Each Equation

Numerical method for u

The numerical solver for this equation is the same as what we did for the continuum case.

Numerical method for X_i

There are many numerical methods for solving ODEs among which the classical Runge-Kutta method stands out for its high order of accuracy and is chosen as the solver for our ODEs. The formula for the RK4 method can be described as follows

$$\begin{aligned} k_1 &= -q^*(t^n, X_i^n), \\ k_2 &= -q^*(t^n + \frac{1}{2}\Delta t, X_i^n + \frac{1}{2}\Delta t k_1), \\ k_3 &= -q^*(t^n + \frac{1}{2}\Delta t, X_i^n + \frac{1}{2}\Delta t k_2), \\ k_4 &= -q^*(t^n + \Delta t, X_i^n + \Delta t k_3), \\ X_i^{n+1} &= X_i^n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned} \tag{3.19}$$

Numerical method for η

After we solve ODEs for the capacity function $X_i(t)$, we will get the hitting time for all companies. Then we sort them in the order that $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_N$, where τ_k is the kth largest hitting time at which the corresponding company exhausts its capacity. Whenever the time reaches the hitting time, there will be a jump in the proportion function η. The formula is as defined in (3.17).

Numerical method for $\bar{p}$

To get the approximate average price solution $\bar{p}^n$, the first thing needed to do is to compute the derivative at point X_i^n of function u^n. This can be achieved by a linear interpolation method. After that we can update the solution $\bar{p}^n$ with the formula described in (3.18).

3.2.4 Summary of Numerical Method for N-player Games

The overall scheme for solving N-player games is described in Algorithm 2. The input is the initial capacity X_i^0 for each company, the terminal condition u^L for the value function, and initial guesses for the proportion and average price functions. The output is the approximate solutions to our N-player games. The tolerance for the stopping criterion is $tol = 10^{-6}$.

Algorithm 2 $(\hat{X}_i, \hat{u}, \hat{\eta}, \hat{\bar{p}}) = N\_player\_solver(X_i^0, u^L, \hat{\eta}^0, \hat{\bar{p}}^0)$

Set: $\hat{\eta}^k = \hat{\eta}^0$, $\hat{\bar{p}}^k = \hat{\bar{p}}^0$.
while $\{(\hat{X}_i^{k+1} - \hat{X}_i^k > tol)$ or $(\hat{u}^{k+1} - \hat{u}^k > tol)$ or $(\hat{\eta}^{k+1} - \hat{\eta}^k > tol)$ or $(\hat{\bar{p}}^{k+1} - \hat{\bar{p}}^k > tol)\}$ **do**

Step 1 Let $\hat{u}^k = \hat{u}^{k+1}$, $\hat{\eta}^k = \hat{\eta}^{k+1}$ and $\hat{\bar{p}}^k = \hat{\bar{p}}^{k+1}$.

Step 2 Using $(\hat{\eta}^k, \hat{\bar{p}}^k)$ to solve the HJB equation for u for all time steps and obtain the approximate solution $\hat{u}^{k+1}$.

Step 3 Using $(X_i^0, \hat{u}^{k+1}, \hat{\eta}^k, \hat{\bar{p}}^k)$ to solve ODEs for all companies in every time step and obtain the approximate solution $\hat{X}_i^{k+1}$.

Step 4 Update the solution to η using the formula in (3.18) and obtain $\hat{\eta}^{k+1}$.

Step 5 Update the solution to $\bar{p}$ using formula in (3.19) and obtain $\hat{\bar{p}}^{k+1}$.

end while
Return: $(\hat{X}_i^{k+1}, \hat{u}^{k+1}, \hat{\eta}^{k+1}, \hat{\bar{p}}^{k+1})$.

Chapter 4

Multigrid Method for HJB Equations

4.1 Multigrid Methods

Multigrid methods are motivated by the analysis of standard iterative techniques such as Gauss-Seidel and Jacobi schemes. These schemes have slow convergence when the system size is large. However, they can reduce high frequency errors very quickly. This observation promoted the development of multigrid methods. A multigrid scheme employs grids of different mesh sizes. Since the error between the approximate solution and the exact solution is smooth after applying standard iterative methods, most of the information will be kept when it is restricted to a coarser grid. The coarse grid problem can be solved to get the solution to the error. Interpolating the error back to the fine grid and adding it to the approximate solution will give a better approximation to the exact solution. This process is called coarse grid correction. It can remove the low frequency error efficiently. In general, the multigrid method combines two parts. At first high frequency errors are reduced by the standard iterative method which is named as smoother. Secondly, low frequency errors are reduced by the coarse grid correction process.

4.1.1 Smoother

Since the Jacobi method is used as the smoother in our multigrid scheme for the HJB equation, we will give an introductory examination for it in this part.

Consider a one-dimensional boundary value problem in the unit interval $x \in [0, 1]$ which

is defined as

$$\begin{aligned} -u_{xx}(x) &= f(x) \quad 0 < x < 1, \\ u(x) &= 0 \qquad\quad x = 0, 1. \end{aligned}$$

The discretization for this equation using finite difference methods with grid size $h = \frac{1}{N}$, $x_i = i \cdot h,\ i = 0, 1, \ldots, N$ is given by

$$\begin{aligned} &\frac{-v_{i-1} + 2v_i - v_{i+1}}{h^2} = f_i, \quad i = 1, 2, \ldots, N-1, \\ &v_0 = v_N = 0, \end{aligned}$$

where v is the approximate solution to the exact solution $u(x)$. Let $\hat{v}^k$ be the kth approximate solution to this problem. The component form for the Jacobi iterative method can be written as

$$\begin{aligned} &\hat{v}_i^{k+1} = \frac{1}{2}(h^2 f_i + \hat{v}_{i-1}^k + \hat{v}_{i+1}^k), \quad i = 1, 2, \ldots, N-1, \\ &\hat{v}_0^{k+1} = \hat{v}_N^{k+1} = 0. \end{aligned}$$

In order to improve the smoothing effect, a damping factor ω is introduced and then a damped-Jacobi method is obtained as

$$\begin{aligned} &\hat{v}_i^{k+1} = (1-\omega)\hat{v}_i^k + \frac{\omega}{2}(h^2 f_i + \hat{v}_{i-1}^k + \hat{v}_{i+1}^k), \quad i = 1, 2, \ldots, N-1, \\ &\hat{v}_0^{k+1} = \hat{v}_N^{k+1} = 0. \end{aligned}$$

4.1.2 Coarse Grid Correction

The problem in Section 4.1.1 can be expressed in the matrix form as $Au = f$ where A is of size $(N-1) \times (N-1)$ and u is the vector form of the exact solution. Let v be an approximation to the exact solution u. The error is defined as $e = u - v$, and the residual of v is $r = f - Av = Au - Av = A(u - v) = Ae$ which is called the residual equation. The error correction is $u = v + e$.

The idea for coarse grid correction is

- Relax on $Au = f$ on the fine grid Ω_h to obtain an approximate solution v_h,
- Compute the residual $r = f - Av_h$,

- Restrict the residual equation to the coarse grid and get $A_H e_H = f_H$,
- Solve the problem on the coarse grid Ω_H to obtain an approximation to errors e_H,
- Interpolate errors e_H back to the fine grid and get e_h,
- Correct the approximate solution on fine grid $v_h \leftarrow v_h + e_h$.

In order to implement this procedure, we need methods for mappings $\Omega_h \to \Omega_H$ and $\Omega_H \to \Omega_h$ which are known as restriction and interpolation respectively. Assuming these two inter-grid transfer operators R and P are given, the coarse grid correction scheme can be described as Algorithm 3.

Algorithm 3 Coarse Grid Correction Scheme

1 Apply smoother α times on $A_h u_h = f_h$ on Ω_h and get an approximate solution v_h.
2 Compute the residual $r_h = f_h - A_h v_h$.
3 Restrict the residual to the coarse grid $r_H = R \cdot r_h$.
4 Get the operator A_H on coarse grid.
5 Solve $A_H e_H = r_H$ on Ω_H.
6 Correct fine grid solution $v_h \leftarrow v_h + P \cdot e_H$.

4.1.3 Two-Grid Cycle

The coarse grid correction procedure removes the low frequency error effectively but not the high frequency error while the smoothing procedure is the opposite. In order to achieve better convergence, it is useful to join these two processes together and this yields the two-grid algorithm. In each iteration, there are three components: pre-smoothing, coarse grid correction and post-smoothing. These are shown in Algorithm 4.

4.1.4 V-Cycle

The two-cycle scheme is not very efficient in practice since the coarse grid problem is still time consuming. It is natural to consider solving the problem on the coarse grid recursively. The idea is to move the multigrid cycle directly towards the coarsest grid and then return it back to the finest gird. This is called the V-cycle scheme. Algorithm 5 gives the general V-cycle procedure. The difference between the V-cycle and the two-cycle lies in the solution

Algorithm 4 $v_h^{k+1} = \text{Two-Grid}(A_h, f_h, v_h^k)$

(1) Pre-smoothing
    Compute v_h^k by applying α iterations of smoothing procedure on $A_h u_h = f_h$ with some initial guess v^0.
(2) Coarse Grid Correction
    Compute the residual: $r_h = f_h - A_h v_h^k$.
    Restrict the residual: $r_H = R \cdot r_h$.
    Solve on coarse grid Ω_H : $A_H e_H = r_H$.
    Correct fine grid solution $v_h^k \leftarrow v_h^k + P \cdot e_H$.
(3) Post-smoothing
    Compute v_h^{k+1} by applying β iterations of smoothing procedure to v_h^k.

Algorithm 5 $v_h^{k+1} = MG(A_h, f_h, v_h^k)$

(1) Pre-smoothing
    Compute v_h^k by applying α iterations of smoothing procedure on $A_h u_h = f_h$ with some initial guess v^0.
(2) Coarse Grid Correction(recursive form)
    Compute the residual: $r_h = f_h - A_h v_h^k$.
    Restrict the residual: $r_H = R \cdot r_h$.
    If Ω_H is the coarsest grid **then**
        Solve $A_H e_H = r_H$.
    Else
        $f_H \leftarrow R \cdot r_h$
        $e_H \leftarrow 0$
        $e_H \leftarrow MV(A_H, f_H, e_H)$
    Correct fine grid solution $v_h^k \leftarrow v_h^k + P \cdot e_H$.
(3) Post-smoothing
    Compute v_h^{k+1} by applying β iterations of smoothing procedure to v_h^k.

of errors on the coarse grid. In the two-cycle scheme, the coarse grid error is computed directly using the residual equation while in the V-cycle, that error is obtained by applying multigrid cycles recursively using zero as the initial guess for the error on the coarse grid.

4.2 Full Approximation Scheme (FAS)

In our MFG problems, we need to solve a non-linear HJB equation which cannot be solved using the standard V-cycle scheme since it is only appropriate for linear problems. Among the literature of multigrid methods, the Full Approximation Scheme (FAS) proposed by Brandt can be used for solving non-linear problems. We are going to study the FAS in this section.

Consider a non-linear problem $N_h(u_h) = f_h$ on Ω_h with the exact solution u_h. Suppose v_h is an approximate solution to u_h after the smoother. The error is $e_h = u_h - v_h$. Since the operator N_h is non-linear, the residual equation $N_h(u_h) - N_h(v_h) = N_h(e_h)$ in this case may not hold. So it is unable to compute e_h using this equality the same as what we have done in the standard multigrid V-cycle.

Consider the usual residual form for this non-linear problem which is given by

$$r_h = f_h - N_h(v_h).$$

Since $f_h = N_h(u_h)$, we have

$$r_h = N_h(u_h) - N_h(v_h). \tag{4.1}$$

Rewrite (4.1) as

$$N_h(u_h) = r_h + N_h(v_h). \tag{4.2}$$

Restrict this equation to the coarse grid and we will get

$$N_H(v_H) = R \cdot r_h + N_H(R \cdot v_h). \tag{4.3}$$

In (4.3), we have a non-linear problem $N_H(v_H) = f_H$ on the coarse grid and the right-hand-side is represented as $f_H = R \cdot r_h + N_H(R \cdot v_h)$. We can solve this coarse grid problem using the relaxation scheme directly and get a solution to v_H. In this way, we will get errors on the coarse grid which can be computed from $e_H = v_H - R \cdot v_h$. Then the coarse grid correction process can be performed to improve the approximate solution v_h. Algorithm 6 illustrates the full FAS process.

Algorithm 6 $v_h^{k+1} = FAS(N_h, f_h, v_h^k)$

(1) Pre-smoothing
    Obtain v_h^k by applying α iterations of smoothing procedure on $A_h u_h = f_h$ with some initial guess v^0.
(2) Coarse Grid Correction
    Compute the residual: $r_h = f_h - N_h(v_h)$.
    Restrict the residual: $r_H = R \cdot r_h$.
    Restrict the fine grid solution: $v_H^k = R \cdot v_h^k$.
    Compute the right-hand-side: $f_H = r_H + N_H(v_H^k)$.
    If Ω_H is the coarsest grid **then**
        Solve $N_H(v_H^k) = f_H$ for $\hat{v}_H^k$.
    Else
        Obtain an improved approximation $\hat{v}_H^k$ using FAS: $\hat{v}_H^k \leftarrow FAS(N_H, f_H, v_H^k)$.
    Compute the correction on coarse grid: $e_H = \hat{v}_H^k - v_H^k$.
    Correct fine grid solution $v_h^k \leftarrow v_h^k + P \cdot e_H$.
(3) Post-smoothing
    Compute v_h^{k+1} by applying β iterations of smoothing procedure to v_h^k.

4.3 Multigrid Method for HJB Equations

In this section, we discuss the components used in our multigrid method for solving the discretized HJB equation in (3.7).

4.3.1 Damped Relaxation Smoother for HJB

As mentioned in Section 4.1, a general multigrid method needs a smoother to reduce the high frequency errors. A damped relaxation scheme is described in this part.

Rewrite our discretized equation (3.7) as

$$u_i^n = u_i^{n+1} + \Delta t \max_{p \geq 0} \left\{ \alpha_i^n u_{i-1}^n + \beta_i^n u_{i+1}^n - \gamma_i^n u_i^n + A_i^n p \right\}. \tag{4.4}$$

Since u_i^{n+1} and u_i^n do not depend on the control p, rearranging (4.4) gives

$$0 = \max_{p \geq 0} \left\{ \Delta t \big(\alpha_i^n u_{i-1}^n + \beta_i^n u_{i+1}^n + A_i^n p \big) - (1 + \Delta t \gamma_i^n) u_i^n + u_i^{n+1} \right\}. \tag{4.5}$$

Note that γ_i^n is non-negative, so (4.5) can be rewritten as

$$0 = \max_{p\geq 0}\left\{(1+\Delta t\gamma_i^n)\Big[\frac{\Delta t(\alpha_i^n u_{i-1}^n + \beta_i^n u_{i+1}^n + A_i^n p) + u_i^{n+1}}{1+\Delta t\gamma_i^n} - u_i^n\Big]\right\}. \tag{4.6}$$

Let $\hat{u}^k$ be the kth estimate for u^n. Then a relaxation scheme can be obtained as,

$$\hat{u}_i^{k+1} = \max_{p\geq 0}\left\{\frac{\Delta t(\alpha_i^n \hat{u}_{i-1}^k + \beta_i^n \hat{u}_{i+1}^k + A_i^n p) + u_i^{n+1}}{1+\Delta t\gamma_i^n}\right\}. \tag{4.7}$$

To improve the smoothing effect, a damping factor to the relaxation scheme is introduced and it gives the following damped-relaxation smoother,

$$\hat{u}_i^{k+1} = (1-\omega)\hat{u}_i^k + \omega\max_{p\geq 0}\left\{\frac{\Delta t(\alpha_i^n \hat{u}_{i-1}^k + \beta_i^n \hat{u}_{i+1}^k + A_i^n p) + u_i^{n+1}}{1+\Delta t\gamma_i^n}\right\}. \tag{4.8}$$

This relaxation scheme is used to solve the problem on the coarsest grid in our multigrid methods.

4.3.2 Interpolation and Restriction Operators

As defined before, the restriction operator R transfers vectors from the fine grid to the coarse grid while the back procedure is through an interpolation matrix P. The choice of restriction and interpolation operators relies on the choice of the coarse grid. In our multigrid method for solving the HJB equation, we focus on standard coarsening which doubles the grid size h in every direction. The direct discretization of the system operator is used on the coarse grid. Linear interpolation and restriction operators are applied to our coarse grid correction process.

We start with the interpolation matrix P which can be regarded as a linear operator from $\mathbb{R}^{N/2-1} \to \mathbb{R}^{N-1}$, where $N+1$ is the number of grid points on the fine grid. For values on the coarse grid we map them unchanged to the fine grid. For values on the fine grid but not on the coarse grid we use the average of their coarse grid neighbors. Let v^h, v^{2h} be vectors defined on the fine grid and coarse grid respectively. For example, for $N = 8$, this operation can be described as

$$\begin{pmatrix} 1/2 & & \\ 1 & & \\ 1/2 & 1/2 & \\ & 1 & \\ & 1/2 & 1/2 \\ & & 1 \\ & & 1/2 \end{pmatrix} \begin{pmatrix} v_1^{2h} \\ v_2^{2h} \\ v_3^{2h} \end{pmatrix} = \begin{pmatrix} v_1^h \\ v_2^h \\ v_3^h \\ v_4^h \\ v_5^h \\ v_6^h \\ v_7^h \end{pmatrix}, \tag{4.9}$$

where the first 7 by 3 matrix is our interpolation operator P, v_i^{2h}, $i \in \{1,2,3\}$ is the ith value on the coarse grid Ω_H and v_j^h, $j \in \{1,2,3,4,5,6,7\}$ is the jth element on the fine grid Ω_h.

Restriction is the mapping from the fine grid to the coarse grid $R : \Omega_h \to \Omega_H$, which is $R \cdot v^h = v^{2h}$. We use the full-weighting method to implement this mapping and we have $v_i^{2h} = \frac{1}{4}(v_{2i-1}^h + 2v_{2i}^h + v_{2i+1}^h)$. For example, when $N = 8$, our restriction operation is as follows

$$\begin{pmatrix} 1/4 & 1/2 & 1/4 & & & & \\ & & 1/4 & 1/2 & 1/4 & & \\ & & & & 1/4 & 1/2 & 1/4 \end{pmatrix} \begin{pmatrix} v_1^h \\ v_2^h \\ v_3^h \\ v_4^h \\ v_5^h \\ v_6^h \\ v_7^h \end{pmatrix} = \begin{pmatrix} v_1^{2h} \\ v_2^{2h} \\ v_3^{2h} \end{pmatrix}, \tag{4.10}$$

where the first 3 by 7 matrix is our restriction matrix.

As we can see, for the 1D linear interpolation and full-weighting restriction, operators P and R are related. In our case, we have $R = \frac{1}{2}P^T$, where P^T is the transpose of matrix P.

Chapter 5

Numerical Results

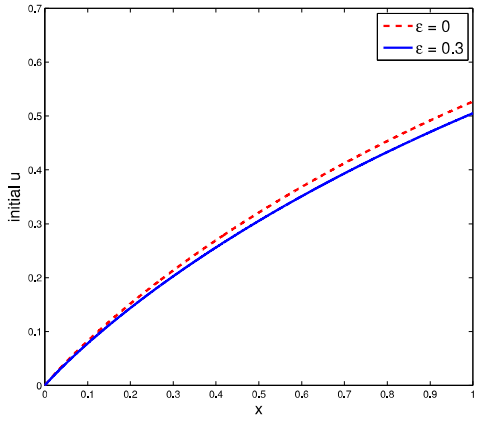
Our numerical algorithms for dynamic mean field games have been described explicitly in the last two chapters. Numerical results obtained from those schemes will be shown in this chapter. We can see our numerical methods work well for MFG problems and the multigrid method for solving the HJB equation is efficient. Results illustrate some properties of the proportion function $\eta(t)$ and the average equilibrium price.

5.1 Numerical Results for Dynamic Continuum Mean Field Games

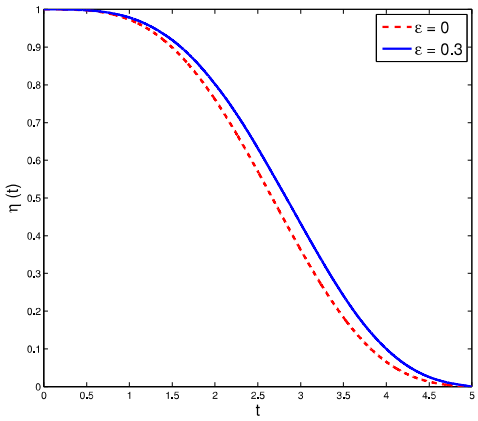
As we mentioned before, the framework for continuum mean field games is the basis of our numerical analysis. Firstly we are going to solve dynamic continuum mean field games both for the deterministic and stochastic cases.

5.1.1 Deterministic Bertrand Mean Field Games

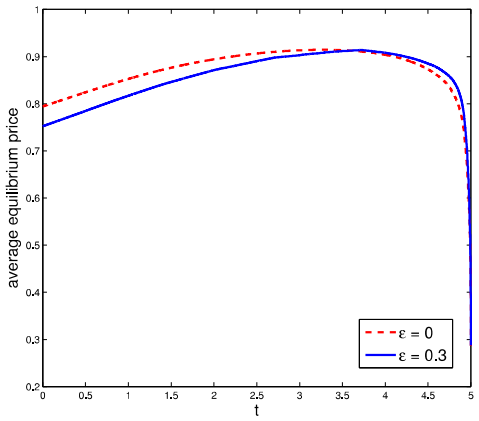
Consider deterministic Bertrand mean field games in which demands are not subject to random fluctuations and $\sigma = 0$. This problem is solved using the numerical method described in Section 3.1. We show results for both cases that $\epsilon = 0.0$ and $\epsilon = 0.3$. When $\epsilon = 0.0$, there is not any competition among players and companies are monopolists in their own markets. When $\epsilon = 0.3$, there exists some competition and the degree is measured by ϵ in the market. The discount rate r in the value function is set to be 0.2. A beta distribution with shape parameters $\alpha = 2, \beta = 4$ is assumed for the initial capacity function $M(x)$.



(a) Initial value $u(0, x)$



(b) Proportion $\eta(t)$



(c) Average equilibrium price $\bar{p}(x)$

Figure 5.1: Numerical results for deterministic Bertrand mean field games

Figure 5.1 illustrates the result plots for the initial value function $u(0, x)$, the proportion of remaining companies $\eta(t)$ and the average equilibrium price $\bar{p}(t)$. As the plot (b) shows, when there is competition among companies, the proportion of remaining players decreases more slowly than the case that $\epsilon = 0.0$. This means players will slow down their speeds in exhausting reserves when faced with competition. In plot (c), the average equilibrium price for the monopoly case is generally higher than that of the competitive market. This is reasonable since consumers tend to buy products at low prices when sellers offer substitute goods in the market.

5.1.2 Stochastic Bertrand Mean Field Games

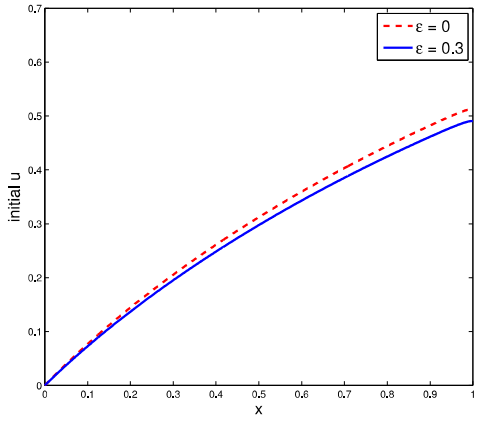
In this section, we give results for stochastic Bertrand mean field games. In this case $\sigma > 0$ which means demands are also subject to random fluctuations.

Figure 5.2 shows a group of numerical results for stochastic Bertrand mean field games with $\sigma = 0.1$. Comparing with result plots we have shown in Figure 5.1, we can see that when $\sigma > 0$, the proportion of remaining companies $\eta(t)$ decreases more rapidly and hence the time duration of the game is shorter. For the average equilibrium price, we notice that the existence of the stochastic term in the demands lowers the amplitude of $\bar{p}(t)$.

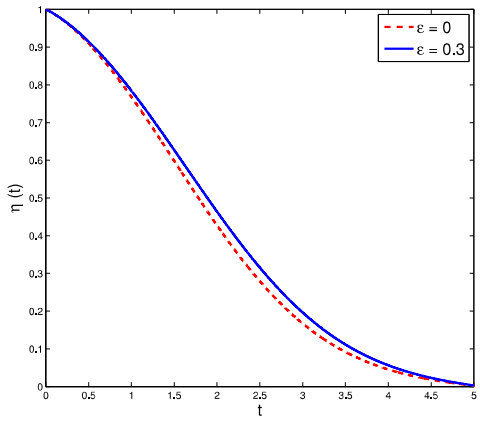
5.2 Numerical Results for N-Player Games

Now we are going to apply our numerical algorithm to solve a 10 player game problem. These 10 players are set to be with initial capacities as $(0.08, 0.14, 0.19, 0.24, 0.29, 0.34, 0.39, 0.45, 0.53, 0.66)$. We show results when $\epsilon = 0.3$.

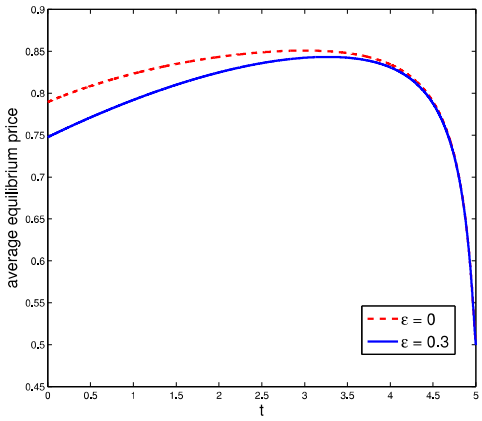
Figure 5.3 shows numerical results for this problem. In plot (a), there are ten lines representing capacity trajectories for ten players respectively. Each line starts with the initial capacity and goes down with time. From this plot, we can get the hitting time for each player and small players exhaust their capacities earlier. As plot (c) shows, whenever one company exhausts its reserves, there will be a jump in the proportion function $\eta(t)$. There are ten jumps in plot (c).



(a) Initial value $u(0, x)$

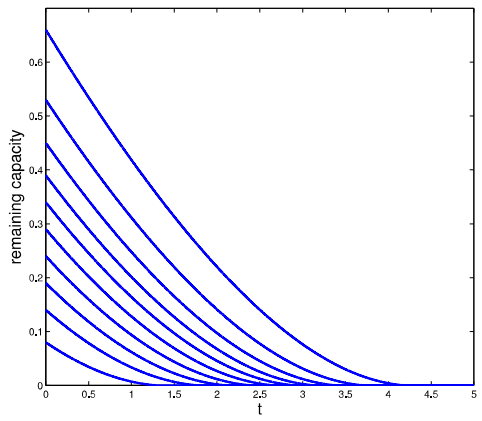


(b) Proportion $\eta(t)$

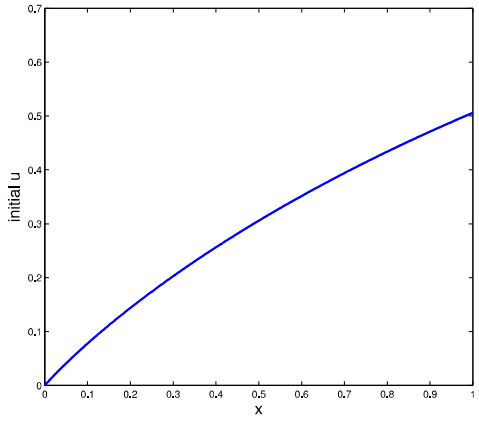


(c) Average equilibrium price $\bar{p}(x)$

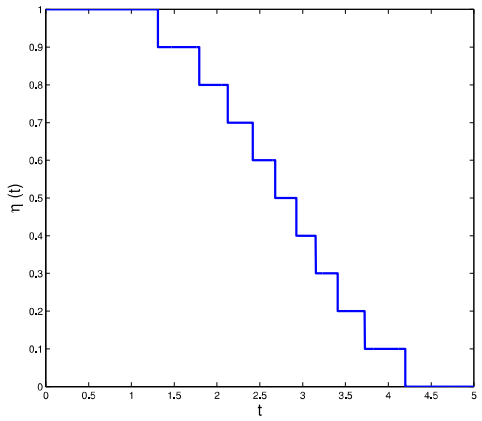
Figure 5.2: Numerical results for stochastic Bertrand mean field games



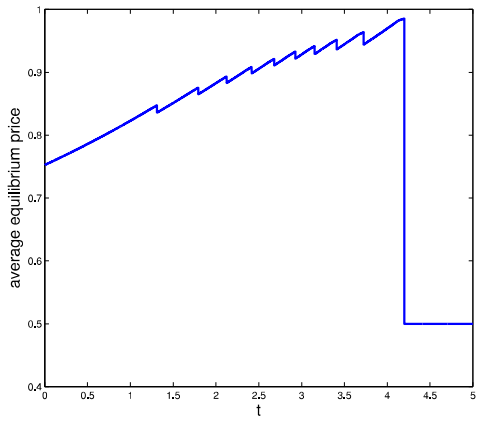
(a) Capacity trajectories $X(t)$



(b) Initial value $u(0, x)$



(c) Proportion $\eta(t)$



(d) Average equilibrium price $\bar{p}(x)$

Figure 5.3: Numerical results for 10 player games

5.3 Convergence Comparison Between Multigrid Method and Relaxation Scheme for Solving HJB Equations

As we have mentioned before, the multigrid method is more efficient than the relaxation scheme in solving the HJB equations. Table 5.1 shows the convergence comparison between the relaxation scheme and the multigrid method with different grid sizes. We can see as the grid size decreases, the number of iterations that the relaxation scheme needs to converge increases very fast while the convergence rate of the multigrid method is not that sensitive to the grid size.

Table 5.1: Convergence comparison

Grid size	Relaxation	Multigrid
$\frac{1}{8}$	28	4
$\frac{1}{16}$	47	5
$\frac{1}{32}$	80	6
$\frac{1}{64}$	141	9
$\frac{1}{128}$	250	13

Chapter 6

Conclusion

In this essay, we studied dynamic Bertrand mean field games with exhaustible capacities which can be formulated into a coupled PDE system consisting of a backward HJB equation and a forward Kolmogorov equation. We used the framework of continuum mean field games to approximate discrete N-player games.

In order to examine effects of the competition on mean field games, we applied an iterative method to solve dynamic Bertrand mean field games. For the value function, in P. Chan and R. Sircar's work [7], they solved the quadratic equation in (2.26) with methods of lines, while in our work, a multigrid method with Full Approximation Scheme was proposed to solve the backward HJB equation in (2.25) directly. The forward Kolmogorov equation was solved by a relaxation scheme.

From numerical results, we can see that when there is some competition in the market players will exhaust their capacities more slowly and the time length of the game decreases. The existence of the competition will also reduce the amplitude of the average equilibrium price. In stochastic games, capacities of companies will be exhausted more quickly than the deterministic case. The efficiency of the multigrid method in terms of the convergence rate was illustrated in Section 5.3.

In our mathematical analysis, we have used linear demands to model mean field games. In the future work, some more complicated models for the demand function may need to be developed and used to achieve better practical meanings. In the numerical computation, some methods with higher accuracy orders can be applied to further improve the convergence rate. For example, when computing the integration, the matlab built-in function "trapz" is only first order accurate, so more accurate methods may help speed up the convergence for the numerical methods.

References

[1] T. Basar, G. J. Olsder, and Clsder, *Dynamic noncooperative game theory*, vol. 200, SIAM, 1995.

[2] J. Bather, *A continuous time inventory model*, Journal of Applied Probability, 3 (1966), pp. 538–549.

[3] A. Bensoussan and J. Frehse, *Nonlinear elliptic systems in stochastic game theory*, SFB 72, 1982.

[4] ——, *Regularity results for nonlinear elliptic systems and applications*, Springer, 2002.

[5] J. Bertrand, *Théorie mathématique de la richesse sociale*, Journal des Savants, 67 (1883), pp. 499–508.

[6] E. Chamberlin, *Theory of monopolistic competition*, (1948).

[7] P. Chan and R. Sircar, *Bertrand & Cournot Mean Field Games*, Available at SSRN 2387226, (2014).

[8] A. A. Cournot, *Recherches sur les principes mathematiques de la theorie des richesses, paris 1838*, English transl. by NT Bacon under the title Researches into the Mathematical Principles of the Theory of Wealth, New York, (1897).

[9] J. M. de Bornier, *The Cournot-Bertrand Debate: A Historical Perspective*, History of Political Economy, 24 (1992), pp. 623–656.

[10] E. Dockner, *Differential games in economics and management science*, Cambridge University Press, 2000.

[11] F. Y. Edgeworth, *The pure theory of taxation*, The Economic Journal, 7 (1897), pp. 46–70.

[12] J. Engwerda, *LQ dynamic optimization and differential games*, John Wiley & Sons, 2005.

[13] A. Friedman, *Differential games*, Courier Dover Publications, 2013.

[14] J. Friedman, *Oligopoly theory*, CUP Archive, 1983.

[15] D. Han and J. W. Wan, *Multigrid Methods for Second Order Hamilton–Jacobi–Bellman and Hamilton–Jacobi–Bellman–Isaacs Equations*, SIAM Journal on Scientific Computing, 35 (2013), pp. S323–S344.

[16] B. Jun and X. Vives, *Strategic incentives in dynamic duopoly*, Journal of Economic Theory, 116 (2004), pp. 249–281.

[17] J.-M. Lasry and P.-L. Lions, *Mean field games*, Japanese Journal of Mathematics, 2 (2007), pp. 229–260.

[18] A. Ledvina and R. Sircar, *Dynamic Bertrand oligopoly*, Applied Mathematics & Optimization, 63 (2011), pp. 11–44.

[19] X. Vives, *Oligopoly pricing: old ideas and new tools*, MIT press, 2001.