Ao. Prof. Dr. Gernot Tragler

DIPLOMARBEIT

Optimal Dynamic Management of the Population Mix

Ausgeführt am Institut für
WIRTSCHAFTSMATHEMATIK
der Technischen Universität Wien

unter der Anleitung von
Ao. Prof. Dr. Gernot Tragler

durch
Masoumeh Moridian, BSc
Universumstraße 38/11, 1200 Wien

Wien, am 12. Dezember 2013

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Wien, am 12. Dezember 2013

____

Verfasserin

Masoumeh Moridian, BSc
Universumstraße 38/11
1200 Wien
m.moridian@eypaustria.org

Abstract This thesis deals with a dynamic optimization model aiming at deconcentrating poverty. The core approach is moving poor families to middle-class areas, which may induce middle-class flight. A reasonable model formulation requires that many different parameters have to be taken into consideration. The problem is analyzed by methods of optimal control theory using the MATLAB® toolbox OCMat developed by Dieter Grass from the research unit for Operations Research and Control Systems at the Vienna University of Technology. The basic model as described in previous work will be extended in order to see how the equilibria and the optimal solutions depend on the model assumptions. Furthermore, the case of a so-called DNSS curve will be analyzed.

Keywords Dynamic segregation · Nonlinear optimal control · OCMat Toolbox · Sensitivity analysis · DNSS point

Danksagung

Ich möchte mich bei Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Gernot Tragler für die fachkundliche Unterstützung, während des Entstehungsprozesses dieser Diplomarbeit, und für die Bereitstellung vieler nützlicher Materialien bedanken.

Weiters möchte ich besonderen Dank Projektass.(FWF) Mag.rer.soc.oec. Dr.rer.soc.oec. Andrea Seidl aussprechen, die mich bei technischen Problemen stets unterstützt hat und immer Zeit für mich hatte, wenn es Schwierigkeiten beim Programmieren der mathematischen Modelle gab.

Am meisten jedoch verdanke ich meinen Eltern, Mag.pharm. Afsaneh Hamrah und Mohammad Hossein Moridian, denen ich hiermit diese Arbeit widmen möchte. Sie haben all die Jahre an mich geglaubt und mir meinen Werdegang ermöglicht. Ich schätze mich sehr glücklich Eltern zu haben, die mich stets unterstützen und immer für mich da sind.

Contents

CHAPTER 1

Introduction

This thesis is a continuation of the thesis by Reka Horvath (2011). In particular, the first five chapters of this thesis heavily rely on the thesis by Horvath (2011) and mostly are taken directly from there. The reason for repeating parts of her work is to make my thesis understandable by itself.

This work deals with the problem faced by a social planner who wants to include a stream of poor families into a middle-class area without evoking middle-class flight. Placing too many poor families in a short time would induce current residents to emigrate and even deter other affluent residents from moving in. Both possibilities would reduce the tax base of the community to which the poor families are relocated, and that is counterproductive. But placing too few marginalized families squanders the opportunity to use the resources of the community to help to assimilate poor families into middle class.

This problem has very concrete, practical motivations. It has received considerable attention in academic research including the analysis of group effects, social interactions and networks, in particular with respect to the design of efficient social policies. However, this kind of problem has rarely been addressed with powerful analytic methods such as optimal control applied in the present work.

Social interactions are mathematically closely associated with non-linearities and multiple equilibria. The existence of multiple equilibria is related to the existence of so-called Skiba or DNSS points. Multiplicity means that for given initial states there exist multiple optimal solutions; thus the decision-maker is indifferent about which option to choose, i.e., at such a threshold different optimal paths exist. Small movements away from the threshold typical resolve the indifference and lead to a unique optimal solution. Sethi (1977, 1979), Skiba (1978) and Dechert and Nishimura (1983) explored these points of indifference for the first time when they considered a special class of optimal control problems. In recognition of their studies these points of indifference are denoted as DNSS (Dechert-Nishimura-Sethi-Skiba) or simply Skiba points.

The following work analyzes the strategy of a housing mobility program, which places poor families into middle-class areas, by using methods of optimal control theory. The underlying premise is that poor families can do better on a variety of social, health, education, and economic indicators if they have the opportunity to choose good-quality housing in more-affluent destination communities. The fundamental management question is, how best could such a strategy look like?

As mentioned above, this thesis is a continuation of a recent work from Reka Horvath (2011) who wrote her diploma thesis on the same topic, analyzing a two-state optimal control model. First of all, I will provide the same overview of the recent problems of segregation in the USA that she provided. Then the formulation of the mathematical model will be discussed, first with one variable and then in terms of a two-state optimal control model. I will provide Reka Horvath's findings about the two-state model and afterwards continue with extensions of the original model, which is also described in Grass and Tragler (2010). The extensions will be named the "α Model" and the "β Model" according to the new parameters introduced via the extensions.

CHAPTER 2

Policy Context: The USA in Words and Figures

According to the United States Census Bureau (2011b) the number of inhabitants of the US expands every twelve seconds by one person. It makes the United States to the third largest country by population with about 308 million people.

This rather high number of people living in the U.S. is not so much caused by a birthrate but rather by a large-scale immigration from many countries. The birthrate is 30% under the world average, which is still higher than that of most of the European countries. One person immigrates to the country every 43 seconds as per United States Census Bureau (2011b), so the United States are one of the world's most ethnically diverse and multicultural nations.

One of the key problems concerning this kind of expansion is the slow process of assimilation of the immigrants. Social inhomogeneity accompanies unemployment and delinquency and it breeds ethnical segregation followed by urban decay. The United States Census Bureau estimates the number of illegal immigrants at about 11.2 million in 2010. The population growth of Hispanic Americans provides the major demographic trend.

The Annual Estimates of the the United States Census Bureau (2011a) provides the following composition of the US population in 2009:

- 79.6% White
- 12.9% African American
- 4.6% Asian American

- 1.0% American Indian and Alaskan Native
- 0.2% Native Hawaiian and Other Pacific Islander
- 1.7% Multiracial

Additionally this study assumes 15.4% of the whole population to be Hispanic, which means 46.9 million people. Hence White Americans are the largest racial group, African Americans are the nation's largest racial minority and Asian Americans are the country's second largest racial minority. Since 1998, China, India, and the Philippines have been in the top four sending countries every year. According to the United States Census Bureau, about 80% of Americans live in urban areas, including suburbs. The "Population Estimates" of the Bureau (2009) specify nine cities with more than one million residents. The biggest metropolises are New York, Los Angeles, Chicago and Houston city with more than two million inhabitants. However, this expansion is associated with large-scale unemployment, where according to the Bureau of Labor Statistics (2011) the average rate amounts to 9.1%. Heavily affected are teenagers by an unemployment of 25.4%, furthermore African Americans by 16.7% and Hispanics by 13.3%, as per the Economic of Labor Statistics (August, 2011). By comparison, as per Die Presse (August 31, 2011), the European Union records an unemployment of 10% with Austria's at 3.7%. The teenager-unemployment in Austria is added up to 7.8%.

According to DeNavas-Walt et al. (2008) the United States denotes the greatest income inequality among developed nations. This report demonstrates also the varying level of income in different states. Maryland has the highest income added up to $68,080 and Mississippi the lowest one by $36,338. Furthermore it sheds light on the American poverty status. In 2008, 13.2% of all Americans lived in poverty, which included more than 30 million people. The harmful effects of high-poverty areas are thought to be especially severe for children whose behavior and prospects may be particularly susceptible to a number of neighborhood characteristics, such as peer group influences, school quality, and the availability of supervised after school activities.

One possibility to reduce destitution is deconcentration of poverty, e.g., via housing mobility programs. By means of dynamic optimization models, this work examines the problem faced by a social planner who wants to integrate poor families into middle-class neighborhoods faced by segregation without inducing "middle-class-flight"[1].

[1]Middle-class-flight is a demographic and sociological term denoting the trend when middle-class people flee desegregated communities due to anxiety of accustomed social standards.

However, one central question is whether flight is driven more by the current inflow of poor immigrants or by their accumulation over time. On that point, there appears to be some reasons to believe it is the current inflow (Ellen, 2000). Charles T. Clotfelter[2] is of the opinion that the Brown v. Board of Education (1954)[3] decision of the Supreme Court - ordering the abrogation of racial segregation of public schools - was and remains the major factor actuating the flight of white Americans from mixed-race communities (Clotfelter, 2004). It is worth to mention, however, that the complexity of problems caused by racial segregation has been a frequently discussed subject at least since the Declaration of Independence, July 4, 1776.

Already, 40 years ago Thomas Schelling[4] has already analyzed this segregative behavior of communities (Schelling, 1971). He showed in his segregation model that a small preference for one's neighbors to be of the same color could lead to total segregation. He used coins on graph paper to demonstrate his theory by placing pennies and nickels in different patterns on the "board" and then moving them one by one if they were in an "unhappy" situation.
The rule, this model operates on, is that for every colored cell, if greater than 33% of the adjacent cells are of a different color, the cell moves to another randomly selected cell. Furthermore, the systemic effects are found to be overwhelming: there is no simple correspondence of individual incentive to collective results. Schelling deduced an "exaggerated separation and patterning result from the dynamics of movement. Inferences about individual motives can usually not be drawn from aggregate patterns" (Schelling, 1971). It is still a powerful example of an "invisible-hand" explanation.

Over the past 10 years the US government has placed an increased emphasis on anti-poverty programs via public housing developments. "The Moving to Opportunity for Fair Housing" (MTO) program directed by HUD (U.S. Department of Housing and Urban Development, 1999) is one of these experimental housing mobility programs (Elhassan et al., 1999). According to the "Moving to Opportunity Interim Impacts

[2]Professor of Public Policy Studies and Professor of Economics and Law at Duke University.

[3]Supreme Court of the United States: Full name of the case: Oliver Brown et al. v. Board of Education of Topeka et al.

[4]Thomas Crombie Schelling (born April 14, 1921) is an American economist and Professor of Foreign Affairs, National Security, Nuclear Strategy, and Arms Control at the School of Public Policy at University of Maryland, College Park. He received the Nobel Prize in Economic Sciences 2005 "for having enhanced our understanding of conflict and cooperation through game-theory analysis".

Evaluation" Report, MTO was designed to answer questions about what happens when very poor families have the chance to move out of subsidized housing in the poorest neighborhoods of five very large American cities, namely Baltimore, Boston, Chicago, Los Angeles, and New York. MTO was a demonstration program: its unique approach combined tenant-based housing vouchers with location restrictions and housing counseling.

The participant families had to live in public housing or private assisted housing in areas of the central cities with very high poverty rates (40% or more), have very low incomes, and have children under 18 years. The mean poverty rate of baseline locations was, in fact, higher than 56%. The experimental Section 8 group was offered housing vouchers that could only be used in low-poverty neighborhoods (where less than 10% of the population was poor, base year 1990) and local counseling agencies helped to find and lease units in qualifying neighborhoods.

The major questions were: What are the impacts of joining the MTO demonstration on household location and on the housing and neighborhood conditions of the participants? What are the impacts of moving to a low-poverty neighborhood on the employment, income, education, health, and social well-being of family members?

A summary assessment of the findings and the impact estimates suggest that: the findings do provide convincing evidence that MTO had real effects on the lives of participating families in the domain of housing conditions and assistance and on the characteristics of the schools attended by their children; there is no convincing evidence of effects on educational performance, employment and earnings, household income, food security, or self-sufficiency.

However, the ability to measure those effects quantitatively is limited. There are a number of reasons to expect that observing the MTO population over a longer period of time may reveal significant program impacts in domains with no mid-term effects. There are strong theoretical reasons why it may take many years for the full effects of neighborhood to manifest themselves. Developmental outcomes such as educational performance almost certainly reflect the cumulative experience of the child from an early age. The analyses found at least modest evidence of increasingly favorable effects over time (Elhassan et al., 1999).

CHAPTER 3

The Model

Caulkins et al. (2005a,b) deal with a simplified one-state model while Grass and Tragler (2010) recently studied the full two-state model for the first time. The model described in what follows is clearly stylized, and many considerations are left out in order to frame an essential and transparent dynamic of the problem.

One of the main factors of the health of a given neighborhood is considered to be the number of middle-class families who live there at time t, denoted by the state variable $X(t)$. The second state variable $Y(t)$ represents the number of poor families in the town. With this additional second state variable one can model explicitly the social advancement of marginal families placed by a formal public program into the middle class, with other words the gradual process by which a family remaining in the neighborhood moves up the socio-economic ladder over time. The key policy variable is the rate at which poor families are situated in the neighborhood, denoted by the control variable $u(t)$[1].

The number of middle-class families, X, changes over time because of to three main influences. First, there are the underlying natural or "uncontrolled" dynamics that would be the case even if there were no external interventions (i.e., $u \equiv 0$). In many ways, housing markets act like other economic markets. So prices adjust to balance supply and demand and the population develops to some optimal city size (Henderson, 1974), so the housing stock is fixed at a size that would in case of normal circumstances support some given population (without loss of generality normalized to be unity, $X = 1$). If the resident population was growing beyond this given normal level ($X > 1$), residents would then flow to less congested middle-class neighborhoods. On

[1]Note that the time argument t will mostly be omitted in the following.

the contrary, if the population falls below that level ($X < 1$), local prices would decline, by this attracting immigration from other, comparable middle-class neighborhoods[2]. To describe this natural adjustment process, the logistic growth curve will be adopted.

The second factor influencing the number of middle-class families is "middle-class flight" which is created by the placement of poor families in the neighborhood. The complexity of the incitement of middle-class flight is enormous. Flight may be provoked not only by immigration into the residential neighborhood but rather by immigration of children into the school district (e.g., Clotfelter, 2001; Fairlie, 2002). Some subgroups appear more likely to flee than others. For instance, Ellen (2000) explains that homeowners are more likely to leave than are renters. Ellen (2000) also argues that families with children are more likely to flee than families without children as they are more concerned about social issues. This comes especially into pratice if the children attend public schools. Furthermore she explains, "whites do not appear to care very much about the proportion of a neighborhood that is African-American, whites do tend to avoid neighborhoods in which the proportion of families who are African-American is increasing (independent of the current size of the minority population)". This is similar to the finding of Betts and Fairlie (2003) in the context of native-born and immigrant population that "for every four immigrants who arrive in public high schools, it is estimated that one native student switches to a private school". The answer to a central question whether flight is driven more by the current inflow of poor immigrants or by their accumulation over time[3] seems to be: the current inflow. So the middle-class flight is assumed to depend primarily as the flow u of marginal families to the stock of current, established families.

The third and final factor influencing changes in the stock of middle-class residents is the social advancement of poor families, which is the rate at which incoming families are "assimilated". The hope is that immersion in a middle-class neighborhood will improve outcomes, including labor market participations and income and educational outcomes for the children, which translate into social opportunity and higher incomes over time. However, in accordance with Mayer and Jencks (1989), there is a possibility that affluent neighbors provoke resentment among the poor over their relative deprivation. A satisfying short-term result such as improving social welfare of the neighborhood is almost impractical. Rather, the work is aimed at the long-term

[2]If the population base of the whole city is changing, i.e., if it is booming or eroding, the neighborhood's normal population density would accordingly change. This analysis assumes the normal population to be constant over time.

[3]Understood relative to the size of the stock of middle-class families.

benefits as mentioned above.

Summing up the effects influencing the population system, Figure 3.1 illustrates the dynamics of the model.

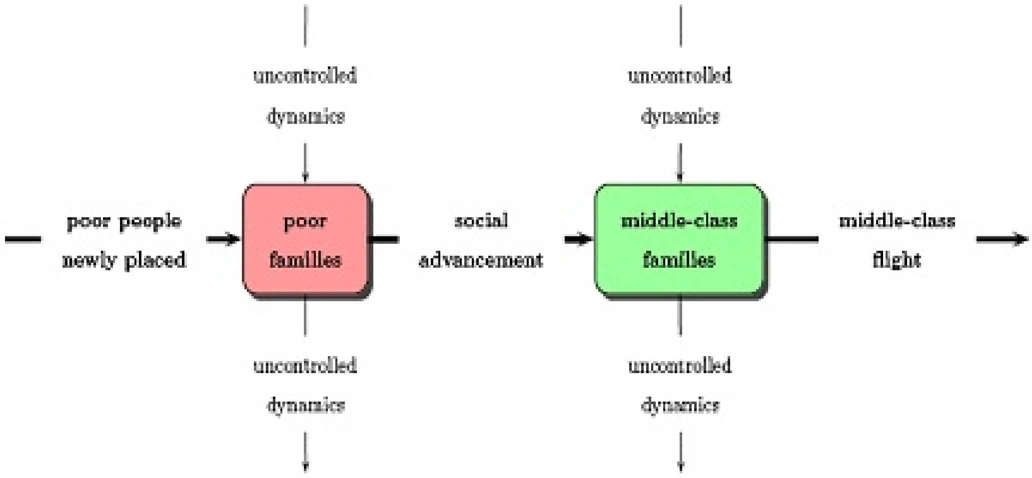

Figure 3.1: Placing the poor to the middle-class neighborhood

After all these observations we get the following dynamic optimization problem with two state variables:

$$\max_{u(\cdot)\geq 0} \int_0^\infty e^{-rt} \left[\rho_X X(t) + \rho_Y Y(t) + \sigma(u(t) - cu(t)^2)\right] dt$$

$$s.t. \quad \dot{X} = own\ dynamics - middle\text{-}class\ flight + social\ advancement$$

$$\dot{Y} = own\ dynamics - social\ advancement + u,$$

where r is the time discount rate, which is non-negative. ρ_X values the presence of established families and ρ_Y the one of marginal families.

Now we take a look at different cases that might occur. If we have $\rho_X = 0$, we have the case that the decision-maker is only concerned about placing as many families as possible. If $\rho_X > 0$, the decision-maker also values directly the presence of established families. So this is in a bit more complex. A reason for the decision maker valuing those families can be the fact that they pay taxes. Here we assume that $\rho_X > \rho_Y$. If we leave this out we can actually get very interesting insights but we won't focus on that for the moment.

We find a weight σ for the control terms in the objective function. By doing this we can easily put more or less emphasis on the contributions from the control variable u relative to those from the state variables X and Y. So there is always a relation between the control variable and the state variables. Here we have quadratic control costs where c is the cost coefficient. Now we have to think about the parameter c and figure out what value makes sense for it. What sometimes we do in Mathematics is leaving out some terms temporarily in order to try out what kind of parameters might fit. Obviously, this is easier if the term is not too complex. This is the reason why we leave out the terms $\rho_X X(t) + \rho_Y Y(t)$ for the moment, so we focus only on the instantaneous part of the objective function, which is $\sigma(u(t) - cu(t)^2)$. In that case we put the total emphasis on the control. We ask ourselves when these control terms are maximized. The answer is, when $u = \frac{1}{2c}$. If we have $c = 2$ we get $u = 0.25$ as the optimal level of the control. That means that we would place one poor family per four middle-class families per year. Placing one poor family per two middle-class families would be far too aggressive and result in less benefits. The only fact that we also have to consider is that all these judgements are most probably tempered by long-run considerations including assimilation and middle-class flight. For further details see Caulkins et al. (2005a,b)

Now we want to explain the own dynamics. We use a very common way that is often mentioned in mathematics which is the logistic growth:

$$\textit{own dynamics} = aX(1 - X).$$

X represents the relative size of the population, and a (b, respectively, for the growth of Y) describes the speed with which the equilibrium population is approached.

The middle-class flight looks as follows:

$$\beta f\left(\frac{u}{X}\right)X,$$

where β is the extent of middle-class flight. Betts and Fairlie (2003) mentioned that one native-born person moved out of the school district for every four immigrants entering ($\beta = 0.25$). The assumption that flight by facing lower-class could possibly be stronger than flight from immigrants suggests even larger values of β. Ellen (2000) argues that a flight coefficient in the range of $0.9 - 1.575$ is reasonable. Taking all this research into consideration, Caulkins et al. (2005a,b) set $\beta = 0.5$ as base case value. For the increasing function $f()$ we set

$$f\left(\frac{u}{X}\right) = \frac{u}{X}.$$

The social advancement term is proportional to Y with the proportionality factor γ, which is the rate of assimilation of poor families into middle-class. The social advancement term is an increasing function of the proportion of those neighbors who are middle-class. One of the important requirements of moving to opportunity programs is that marginal families learn from their more affluent neighbors and adopt the "successful" practices. Through this they also achieve middle-class status. So we can explain the social advancement term as follows:

$$\textit{social advancement} = \gamma Y g(X, Y),$$

with $g()$ increasing in X. $g()$ can for instance have the following form:

$$g(X, Y) = \left(\frac{kX}{kX + Y} \right)^e,$$

where $e > 0$. k is the extent to which the neighborhood was integrated. $k = 1$ stands for random mixing. In this case the proportion of middle-class people to whom a marginal family is exposed equals the proportion of middle class families in the town. If $k < 1$, the proportion of middle-class families seen is less than their factual proportion of the population of the town.

Together, all these reflections suggest the following formulation of the model:

$$\max_{u(\cdot) \geq 0} \int_0^\infty e^{-rt} \left[\rho_X X(t) + \rho_Y Y(t) + \sigma(u(t) - cu(t)^2) \right] dt \tag{3.1}$$

subject to the dynamic state equations

$$\dot{X}(t) = aX(t)(1 - X(t)) - \beta u(t) + \gamma Y(t) \left(\frac{kX(t)}{kX(t) + Y(t)} \right)^e \tag{3.2}$$

$$\dot{Y}(t) = bY(t)(d - Y(t)) - \gamma Y(t) \left(\frac{kX(t)}{kX(t) + Y(t)} \right)^e + u(t) \tag{3.3}$$

with base case parameters presented in Table 3.1.

Table 3.1: Base case model parameters.

Parameter	Value	Description
r	0.05	discount rate
ρ_X	0.02	objective function coefficient on X
ρ_Y	0.01	objective function coefficient on Y
σ	0.01	weight on objective function control terms
c	2	program cost coefficient
a	2	maximal growth rate at $X = 0$
b	2	maximal growth rate at $Y = 0$
d	1	carrying capacity of Y
β	0.5	flight coefficient
γ	0.45	assimilation coefficient
k	1	social integration coefficient
e	1	exponent in the social advancement term

CHAPTER 4

The MATLAB-Toolbox: OCMat

The OCMat Toolbox initiated by Dieter Grass enables an appropriate analysis of optimal control problems using the MATLAB® language[1]. OCMat is mostly used for analysing discounted, autonomous, infinite time horizon models but is also provides extensions to non-autonomous, finite time horizon problems. The numerical method of the toolbox used to solve optimal control problems is based on Pontryagin's Maximum Principle, which establishes the corresponding canonical system. Essentially, solving optimal control problems is translated to the problem of analyzing the canonical system, together with the condition for the initial state and some transversality condition.

In addition to the idea of formulating discounted, autonomous, infinite time horizon optimal control models as BVPs, the occurrence of limit sets (equilibria, limit cycles) of the canonical system as long-run optimal solution was the key argument for Dieter Grass to use a continuation method to anaylze those BVPs. In general, continuation means continuing an already detected solution while varying a model-specific parameter value. Of course, in the context of a BVP the interest is in the majority of cases not only in continuing a solution for varying model parameters but also in the continuation of a solution along varying initial conditions $x(0) = x_0$. Limit sets serve as the first "trivial" solution of an optimal control problem and can be continued in order to derive optimal solutions for arbitrary initial states. The existence of these solutions generated by every continuation step is founded by the implicit function theorem.

One can sum up the main ideas used in OCMat as follows:

- transforming the optimal control problem to a boundary value problem;

[1]MATLAB is a registered trademark of The MathWorks Inc.

- using the technique of continuing an already established solution, which is given by an equilibrium or limit set and

- formulating a so-called asymptotic boundary condition.

More precisely, to introduce the BVP approach, D. Grass starts with the reformulation of an optimal control problem, where it is assumed that the stable manifold of the equilibrium $(\hat{x},\, \hat{\lambda})$ is of dimension n and is the long-run optimal solution. Then, given an initial state $x(0) = x_0 \in \mathbb{R}^n$, a trajectory $(x(\cdot),\, \lambda(\cdot))$ has to be found, which satisfies the ODEs of the canonical system and converges to the equilibrium $(\hat{x},\, \hat{\lambda})$ (Grass et al., 2008, p352). Using the definition of a local stable manifold this can be formulated so that for some T

$$(x(T), \lambda(T)) \in W^S_{loc}(\hat{x}, \hat{\lambda}),$$

or approximating $W^S_{loc}(\hat{x}, \hat{\lambda})$ by its linearization $E^S(\hat{x}, \hat{\lambda})$,

$$(x(T), \lambda(T)) \in E^S(\hat{x}, \hat{\lambda}),$$

which provides the terminal condition. Furthermore, the infinite time horizon is replaced by some finite horizon T. Thus the problem is reduced to a BVP, where initial condition $x(0) = x_0$ and the terminal condition from above is given.

This short summary of the principles of OCMat presents its key aspects, which serve as the basis for a better understanding of the present work. However, the capacity of the toolbox extends the facility of analyzing two-stage optimal control models quickly, reliably and hence very efficiently, and it also enables the calculation of other problem classes, such as multi-stage models and differential games.

For further information see the OCMat webpage http://orcos.tuwien.ac.at/research/ocmat_software, which also provides the slides from lectures by Dieter Grass and an OCMat Manual by Dieter Grass and Andrea Seidl. Further details can be found in Grass et. al (2008, Chapter 8).

The next chapter describes the analysis of the underlying discounted, autonomous optimal control model with a specific focus on using OCMat.

CHAPTER 5

Analysis of the Dynamical System

As this thesis is a continuation of a recent work from Reka Horvath (2011), I will shortly describe her analysis and afterwards continue with modifications of the model. The analysis of the underlying system happens in the usual manner for an optimal dynamic control problem with application of Pontryagin's Maximum Principle. It means in its simplest form that the solution of the control problem is delivered from the solution of the so-called canonical system provided by the maximum principle. OCMat is used for the numerical analysis of the system of nonlinear ODEs.

Before the analysis of an optimal control problem with OCMat can be started, some preparing steps have to be done. In particular, a file describing the state dynamics, objective function, and - possibly - control constraint has to be created and initialised. The initialising process consists of two steps: after the creation of the file, MATLAB files containing default information of the model and MATLAB files necessary for the computation have to be generated. For our model, the content of the file has to have the form:

```
statedynamics=sym('[a*x1*(1-x1)-beta*u1+gamma*x2*(k*x1/
(k*x1+x2))^e;b*x2*(d-x2)-gamma*x2*(k*x1/(k*x1+x2))^e+u1]');
objectivefunction=sym('sigma*(u1-c*u1^2)+rhox*x1+rhoy*x2');
controlconstraint=sym('[u1-lb]');
```

where the name of the file is used as the models' name (e.g., BaseCaseModel.m.) `controlconstraint=sym('[u1-lb]')`means the control constraint is $u > 0$. The file also includes the parameter values. It has to be introduced by the comment %General:

```
%General
```

```
r=0.05;
a=2;
b=2;
c=2;
beta=0.5;
gamma=0.45;
e=1;
rhox=0.02;
rhoy=0.01;
d=1;
k=1;
lb=0;
sigma=0.01;
```

The next step is the initialization of the file by

```
initocmat('BaseCaseModel');
m=ocmodel('BaseCaseModel');
files4model(m);
moveocmatfiles(m);
```

`initocmat` derives and stores important information from the ocmodel; `ocmodel` constructs an ocmodel. The constructor loads the data previously stored during the initialization process. `files4model` creates files for the numerical analysis and `moveocmatfiles` moves the model files from the standard output directory to the standard model directory.

The first step of the analysis is to locate the steady states of the canonical system. These are the intersections of the state- and costate-isoclines. For that purpose, command `calcep(m)` is used, with which one can calculate the equilibria analytically, if the system is not too complex. The toolbox also provides the possibility to solve the equations numerically: `rand(4,10)` means in this special case that for the calculation of the equilibrium consisting of four entries (i.e., two states- and two corresponding costate-values: $X, Y, \lambda_1, \lambda_2$) the numerical calculation starts at ten random initial values. The toolbox checks if some solutions are admissible, i.e., they satisfy possible constraints and are actually zeros of the dynamics, with `b=isadmissible(m,ocEP,opt)`. Negativity of state values is checked with `b=isnegativestate(m,ocEP)`. Also repetitive equilibria can be removed by `ocEP=uniqueoc(ocEP,opt)`. With `ocEP{:}` the user gets the calculated set ($\hat{X}$, $\hat{Y}$, $\hat{\lambda}_1$, $\hat{\lambda}_2$). Summing up all these possibilities, the following compound command can be called for calculating the equilibria of the canonical system:

```
ocEP=calcep(m,rand(4,10),opt);b=isadmissible(m,ocEP,opt);ocEP(~b)=[];
b=isnegativestate(m,ocEP);ocEP(b==1)=[];ocEP=uniqueoc(ocEP,opt);ocEP{:}
```

In her further analysis Reka Horvath (2011) comes to the conclusion that the underlying model with base case parameter values exhibits four candidates for an optimal solution by solving the canonical system, but only one of them serves as an equilibrium in the optimal system. That means, the system has one unique optimal steady state solution ($\hat{X}$, $\hat{Y}$), namely $\hat{X} = 1.0485$, $\hat{Y} = 1.0141$, $\hat{\lambda}_1 = 0.009113$ and $\hat{\lambda}_2 = 0.0049762$. The optimal level of middle-class as well as the optimal level of poor families are slightly above the corresponding carrying capacities (= 1), with $\hat{X}$ being insignificantly greater than $\hat{Y}$ and $\hat{u}$ being some 4% above 0.25 (cf. Grass and Tragler, 2010).

CHAPTER 6

The α Model: Rich People Becoming Poor

The first modification that I analyzed in the course of analyzing the two-state optimal control model was the question on what happens if rich people living in the middle-class area become "poor" in the course of time. So they lose their status of middle-class, which influences the optimal steady state solution that was mentioned above.

Modelling this case looks as follows: we have to subtract the term $\alpha X(t)$ from the $\dot{X}$ equation and add this factor to the $\dot{Y}$ equation. Of course the extent of α plays an important role in this model, that is the reason why I called it the "α" Model. Putting these changes of the model together, we get the following new model:

$$\max_{u(\cdot)\geq 0} \int_0^\infty e^{-rt} \left[\rho_X X(t) + \rho_Y Y(t) + \sigma(u(t) - cu(t)^2)\right] dt \tag{6.1}$$

subject to the dynamic state equations

$$\dot{X}(t) = aX(t)(1 - X(t)) - \beta u(t) + \gamma Y(t) \left(\frac{kX(t)}{kX(t) + Y(t)}\right)^e - \alpha X(t), \tag{6.2}$$

$$\dot{Y}(t) = bY(t)(d - Y(t)) - \gamma Y(t) \left(\frac{kX(t)}{kX(t) + Y(t)}\right)^e + u(t) + \alpha X(t). \tag{6.3}$$

In order to know if we can actually work with this model, we first set $\alpha = 0$ in order to check if we receive the same equilibra as in the original model. So we create the file `NewModel.m` and carry out the same initialisation process as before in order to receive the equilibria using the command `calcep`. Doing this we realise that in the case of $\alpha = 0$ we get the same equilibria and thus the same optimal steady state solution as before, namely $\hat{X} = 1.0485$, $\hat{Y} = 1.0141$, $\hat{\lambda}_1 = 0.009113$ and $\hat{\lambda}_2 = 0.0049762$. We call the equilibria of this model `ocEPn` and the above equilibrium can be recalled by typing `ocEPn{1}`.

6.1 Continuing the Equilibria with Matcont

The next step is to change the parameter α in order to see how this influences the values of X and Y and especially the optimal solution $(\hat{X}, \hat{Y}, \hat{\lambda}_1, \hat{\lambda}_2)$. Before having a look at specific values of α and analyzing those in more detail we want to check what happens with X and Y if α becomes bigger and bigger. So we want to continue the original equilibria. This happens the easiest by using the toolbox Matcont which is also a Matlab toolbox that can be used together with OCMat.

Continuing the equilibria happens with the following command:
`[b1,b2,b3]=contep(n,ocEPn{1},'alpha',opt)` where n stands for `NewModel.m`. Before we do this we need to check if α is increasing or decreasing. This happens with the command `opt.MATCONT.Backward=1` or `opt.MATCONT.Backward=[]`. One cannot say which of these commands is responsible for increasing α. This has to be checked manually by the user. `b1` stands for the equibrium which is continued. After doing this we receive 300 equilibria. This number can also be edited but 300 is the standard amount delivered by the program.

Now that we have continued the original equilibrium we can have a look at the bifurcation diagrams. We first take a look at the two-dimensional diagrams before putting them together in a three-dimensional version.

If we want to see how $\hat{X}$ develops if we increase α we need to type the command `plot(b1(end,:),b1(1,:))`. We will have α on the x-axis and $\hat{X}$ on the y-axis. So in the command that we used above `b1(end,:)` stands for the parameter α which is being increased and `b1(1,:)` stands for $\hat{X}$.
Then we will also have a look at the diagram where $\hat{Y}$ is continued. We use a very similar command: `plot(b1(end,:),b1(2,:))` as the only difference we have is exchanging $\hat{X}$ by $\hat{Y}$.

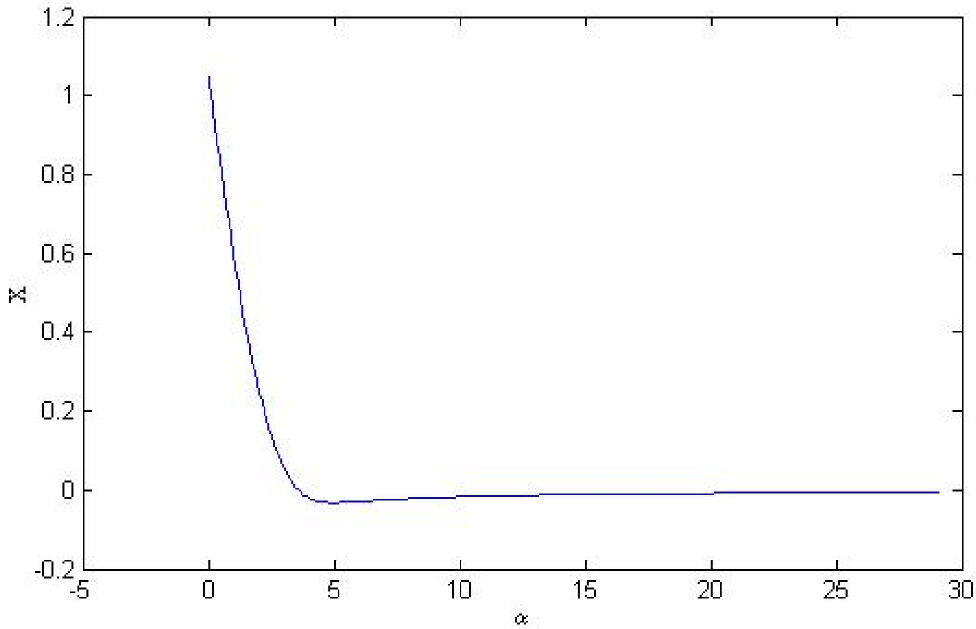

Figure 6.1: Continuing $\hat{X}$ by increasing α

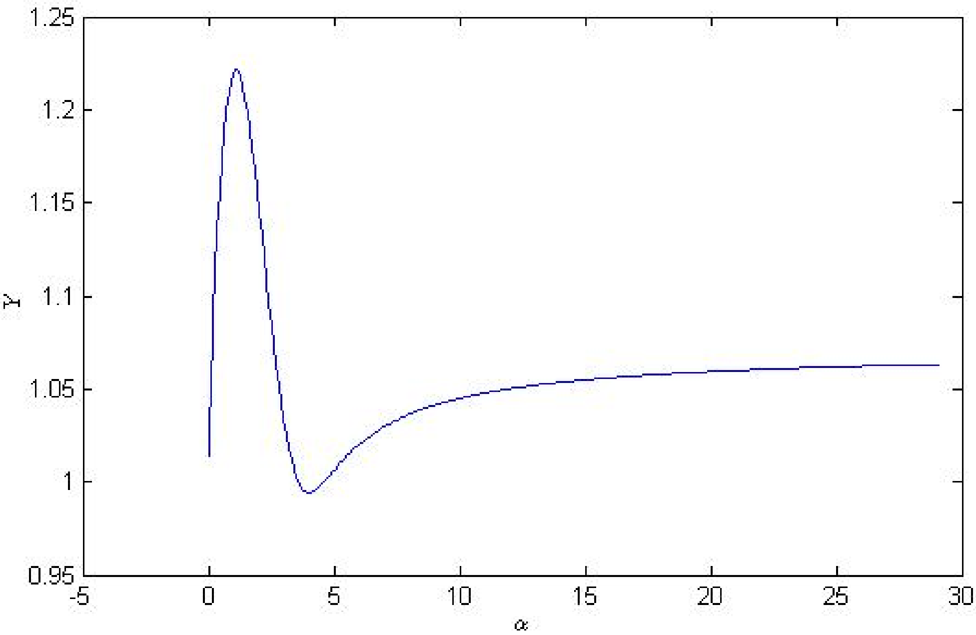

Figure 6.2: Continuing $\hat{Y}$ by increasing α

In addition to the two diagrams we have created so far we will now have a look at the three-dimesional digram where we see the development of $\hat{X}$ and $\hat{Y}$ at the same time. We have to use the command `plot3(b1(1,:),b1(2,:),b1(end,:))`. So we have $\hat{X}$ on the x-axis, $\hat{Y}$ on the y-axis, and α on the z-axis.

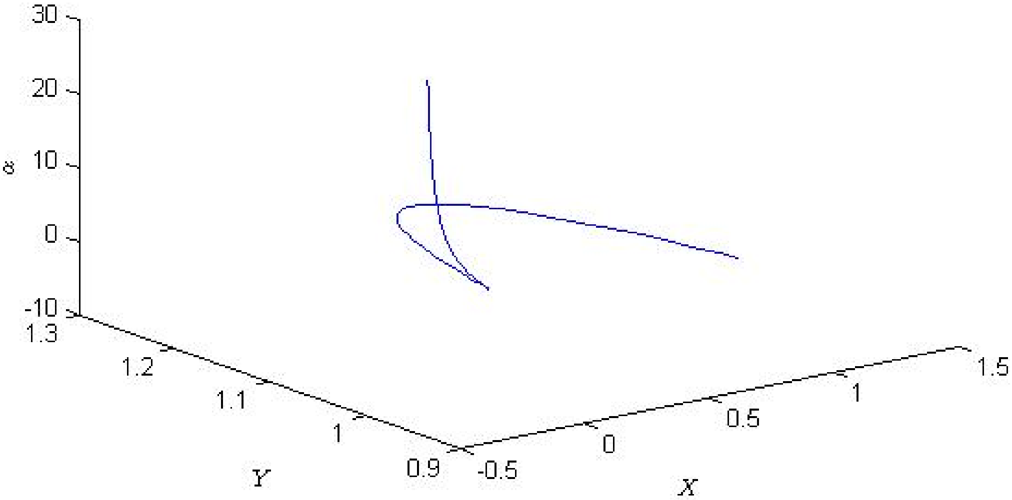

Figure 6.3: Continuing $\hat{X}$ and $\hat{Y}$ by increasing α

Now we would also like to have a bifurcation diagram where $\hat{X}, \hat{Y}$ and the corresponding control $\hat{u}$ are displayed at the same time, all depending on α. First of all we need to get hold of the control $\hat{u}$. We do this by defining the function `controlalpha`:

```
function out = controlalpha(m,b,arcid)

out=[];

for ii=1:size(b,2)
    m=changeparameter(m,'alpha',b(end,ii));
    out(:,end+1)=control(m,b(1:4,ii),arcid);
end
```

This function delivers us the control values of the whole path. So now we have all the inputs that we need for the bifurcation diagram. But first we will have a look at the bifurcation diagram that shows us only the relation of the control $\hat{u}$ and the parameter α.

```
out = controlalpha(n,b1,1)

plot(b1(end,:),out)
```

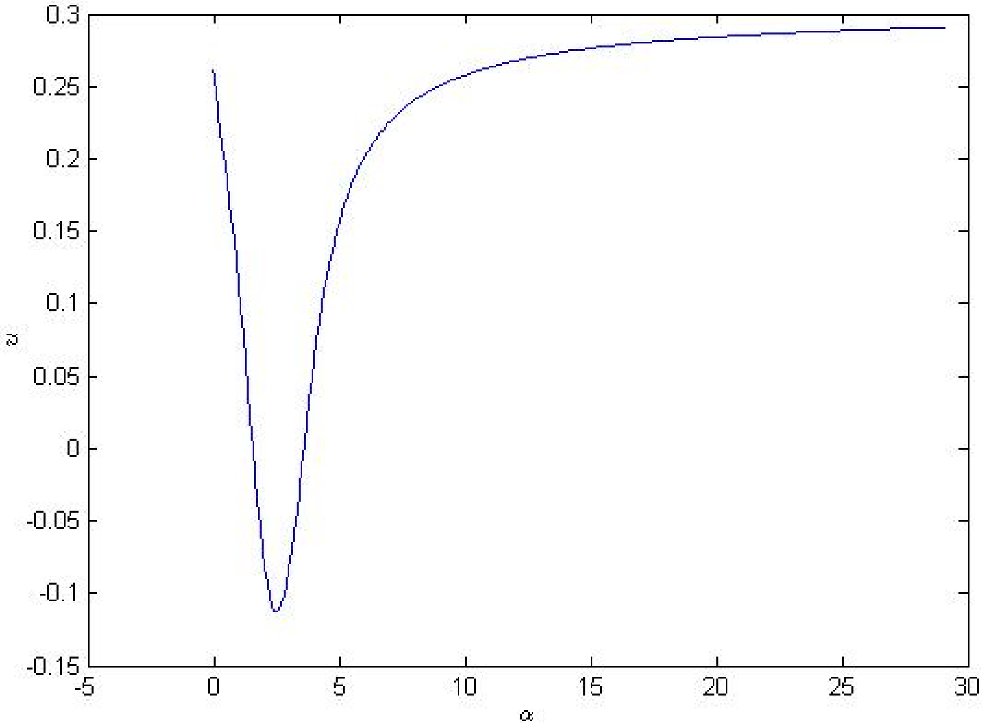

Figure 6.4: Continuing $\hat{u}$ by increasing α

Since we cannot have a four-dimensional diagram we will go back to the two-dimensional version and have all three graphs for $\hat{X}, \hat{Y}$ and $\hat{u}$ in one diagram by using the command `hold all`. So we type the following commands to receive the diagram:

```
plot(b1(end,:),out)
hold all
plot(b1(end,:),b1(1,:))
hold all
plot(b1(end,:),b1(2,:))
hold all
plot(b1(end,:),out)
```

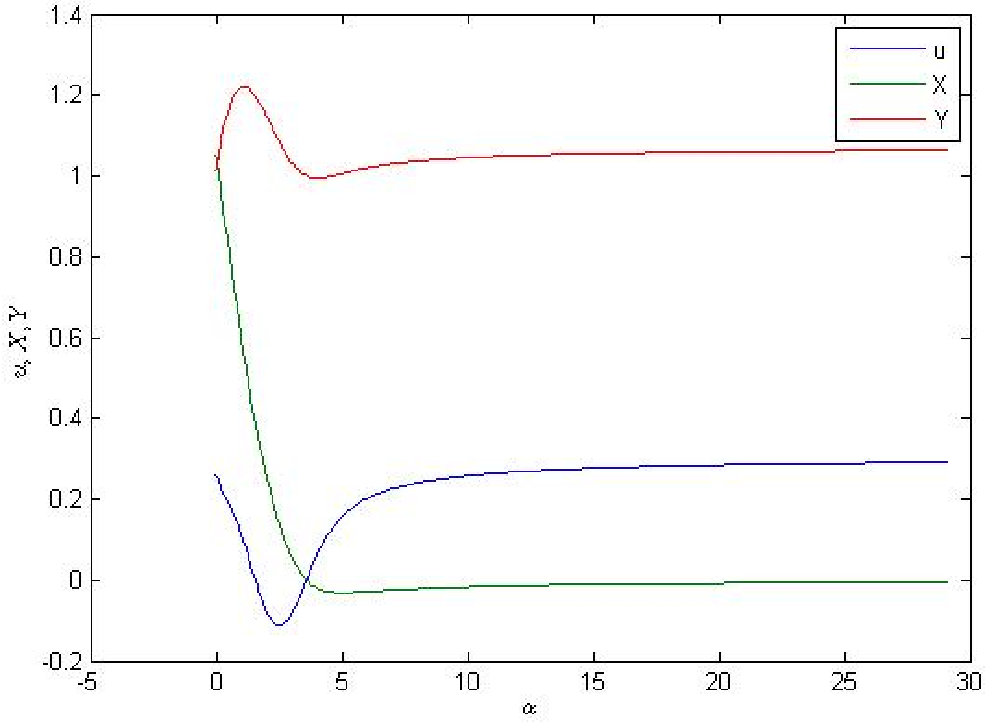

Figure 6.5: Continuing $\hat{X}, \hat{Y}$ and $\hat{u}$ by increasing α

6.2 Changing the Parameter α

In the beginning of the chapter we described a modification of the original model that introduced a new parameter: α. First we examined if the model deliveres the same results by setting $\alpha = 0$. Afterwards we had a closer look at the parameter α and continued the equilibrium by increasing α. Now we will go one step further and take a closer look at particular values of α.

First we need to choose values of α that might be interesting. I have put further analysis of some specific parameter values in the Appendix. First of all we have to make sure that we find ourselves in the admissible area, where $\hat{X}, \hat{Y}$ and $\hat{u}$ are non-negative. Taking a closer look at Figure 6.1, we see that $\hat{X}$ becomes negative for values of α high enough. In Figure 6.2 we see that $\hat{Y}$ is positive throughout, but then Figure 6.4 shows us that there is an interval, in which $\hat{u}$ is negative. To find the exact values we use the command `c1` to have a closer look at the continued equilibrium. There we see that $\hat{X}$ becomes negative starting from index number 45. We type the command `out` to take a look at the values for $\hat{u}$ and see that $\hat{u}$ is negative from index 25 until index 44. This additional restriction means that we can only consider the first 24 equilibria. Fortunately the maximum of the graph of Figure 6.2 lies in that area, so we will first take a look at this value. Unfortunately the minimum lies outside that area so any analysis on that does not make sense. To have another interesting comparison afterwards we will also have a look at parameters of α where $\hat{Y}$ has the same value. You can imagine this by a horizontal line crossing the graph at two different points.

Maximum Value of $\hat{Y}$

First of all we have to find the maximum value of $\hat{Y}$. We do this by typing the following command:

```
find(b1(2,:)==max(b1(2,:)))
```

This command goes through the whole $\hat{Y}$ graph and finds the maximum value, which is at position 19. By typing the command `b1(:,19)` we get the equilibrium $(\hat{X}, \hat{Y}, \hat{\lambda}_1, \hat{\lambda}_2)$ with maximum value for $\hat{Y}$:

```
ans =

    0.5643
    1.2212
    0.0195
    0.0037
    1.0931
```

The last line delivers us the value of α at which $\hat{Y}$ has its maximum value. So we can continue our analysis by using the common procedure from before. First of all we change the parameter of `NewModel.m` from $\alpha = 0$ to $\alpha = 1.0931$ by using the following command:

```
n=changeparameter(n,'alpha',1.0931)
```

Let's have a look at the parameters of the model, considering the change that has just been made:

Table 6.1: α Model parameters.

Parameter	Value	Description
r	0.05	discount rate
ρ_X	0.02	objective function coefficient on X
ρ_Y	0.01	objective function coefficient on Y
σ	0.01	weight on objective function control terms
c	2	program cost coefficient
a	2	maximal growth rate at $X = 0$
b	2	maximal growth rate at $Y = 0$
d	1	carrying capacity of Y
α	1.0931	rich to poor coefficient
β	0.5	flight coefficient
γ	0.45	assimilation coefficient
k	1	social integration coefficient
e	1	exponent in the social advancement term

Although we already received the equibrium we will try calculating it with the original procedure:

```
ocEPn=calcep(n,rand(4,10),opt); c=isadmissible(n,ocEPn,opt);ocEPn(~c)=[];
c=isnegativestate(n,ocEPn);ocEPn(c==1)=[]; ocEPn=uniqueoc(ocEPn,opt); ocEPn{:}
```

which indeed delivers us the above solution:

```
ans =

  dynprimitive object:

      Coordinates:
0.5643
1.2212
0.0195
0.0037

      Arc identifier:  1
      Linearization:   [4x4 double]
```

We can always retrieve this equilibrium by typing `ocEPn{1}`.

Let us also take a look at the stability of this equilibrium. By typing `eig(ocEPn{1})` we get the eigenvalues of the above solution. These are

```
ans =
```

```
-3.0406
-1.2822
 1.3322
 3.0906
```

This exhibits a two-dimensional stable manifold because the number of eigenvalues ξ satisfying $Re\xi < 0$ is two. We had the same case in the original model analyzed by Reka Horvath (2011).

Now we want to receive a phase diagram. We will start from four different points that will all lead to the equilibrium. I will take the same starting points as Gernot Tragler and Dieter Grass used in their paper "Optimal Dynamic Management of the Population Mix" (2010), which are

$(\hat{X}/2, \hat{Y}/2) = (0.2822, 0.6106)$
$(2\hat{X}, 2\hat{Y}) = (1.1286, 2.4424)$
$(\hat{X}/2, 2\hat{Y}) = (0.2822, 2.4424)$
$(2\hat{X}, \hat{Y}/2) = (1.1286, 0.6106)$

Now we have to start a continuation process from the state of one of the above points towards the equilibrium of the maximal value of $\hat{Y}$ and vice versa by using:

```
initStruct1=initoccont('extremal',n,'initpoint',[1 2],[0.2822;0.6106],
ocEPn{1},'IntegrationTime',500);
```

The next step is solving the BVP. The toolbox does this buy using the following command:

```
[soln solnn]=occont(n,initStruct1,opt);
```

Matlab is not able to compute the full path, therefore we have to split it into two parts. So we type the following command:

```
initStruct1=initoccont('extremal',n,'initpoint',[1 2],[0.2822;0.6106],
solnn,'IntegrationTime',500);
```

And then we use again the command `occont`.

```
[solna solnna]=occont(n,initStruct1,opt);
```

For the second and the fourth point there is no need to split the path into two parts. Only the third time requires the same procedure. So we type the following commands:

```
initStruct2=initoccont('extremal',n,'initpoint',[1 2],[1.1286;2.4424],
ocEPn{1},'IntegrationTime',500);

[soln1 solnn1]=occont(n,initStruct2,opt);

initStruct3=initoccont('extremal',n,'initpoint',[1 2],[0.2822;2.4424],
ocEPn{1},'IntegrationTime',500);

[soln2 solnn2]=occont(n,initStruct3,opt);

initStruct3=initoccont('extremal',n,'initpoint',[1 2],[0.2822;2.4424],
solnn2,'IntegrationTime',500);

[soln2a solnn2a]=occont(n,initStruct3,opt);

initStruct4=initoccont('extremal',n,'initpoint',[1 2],[1.1286;0.6106],
ocEPn{1},'IntegrationTime',500);

[soln3 solnn3]=occont(n,initStruct4,opt);
```

Now we want to have a phase diagram with all four paths. By using the command `hold all` we manage to have different graphs in one diagram. So we type the following commands, one by one:

```
plot(solna.dynVar(1,:),solna.dynVar(2,:))
hold all
plot(solna.dynVar(5,:),solna.dynVar(6,:))
hold all
plot(soln1.dynVar(1,:),soln1.dynVar(2,:))
hold all
plot(soln2a.dynVar(1,:),soln2a.dynVar(2,:))
hold all
plot(soln2a.dynVar(5,:),soln2a.dynVar(6,:))
hold all
plot(soln3.dynVar(1,:),soln3.dynVar(2,:))
```

After typing all these commands we get the phase diagram as depicted in Figure 6.6.

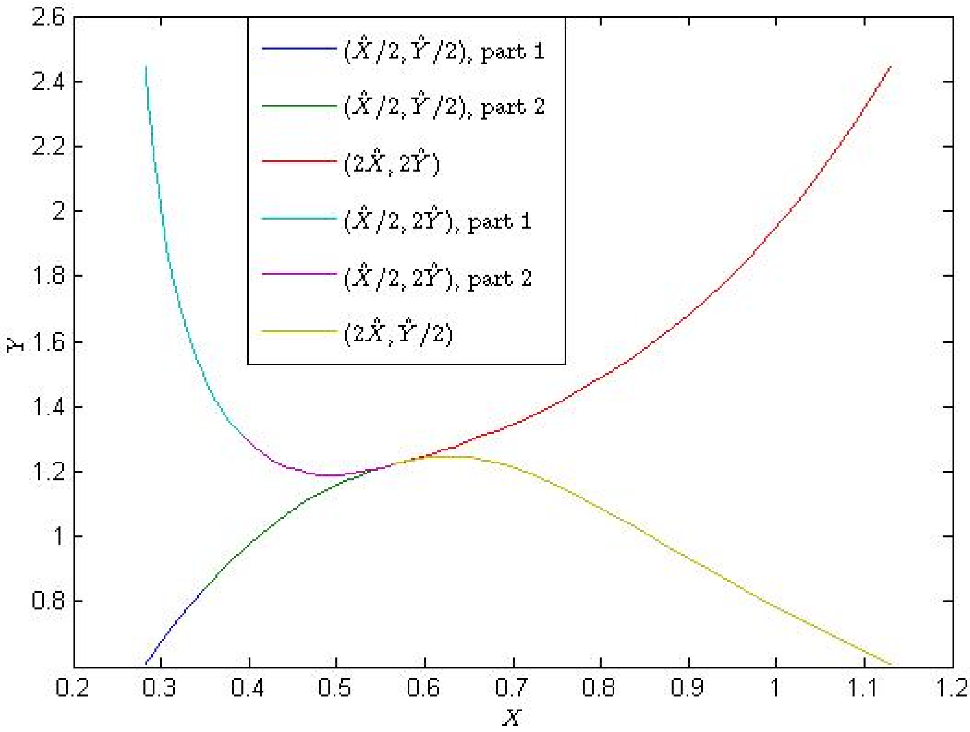

Figure 6.6: Phase diagram for the maximum value of $\hat{Y}$

Two Equal Values of $\hat{Y}$

As we said before we also want to have a look at two different parameter values of α that give us the same value of $\hat{Y}$. So we take a closer look at the output which is delivered by `b1(2,:)`. We find out that there are similar values for instance at the position 15 and 24 of the path, which deliver $\hat{Y} = 1.2002, 1.1986$ for $\alpha = 0.7264, 1.5604$, respectively.

Now we want to receive the phase portraits. We start with $\alpha = 0.7264$. By typing `b1(:,15)` we get the complete equilibrium including α in the last line.

```
ans =

    0.7223
    1.2002
    0.0148
    0.0038
    0.7264
```

Now we can continue our analysis by using the common procedure from before. First of all we change the parameter α of `NewModel.m` from by using the following command:

```
n=changeparameter(n,'alpha',0.7264)
```

Table 6.2 summarizes the parameter values of the model, considering the change that has just been made. Now we will try calculating the above equilibrium with the original

Table 6.2: α Model parameters.

Parameter	Value	Description
r	0.05	discount rate
ρ_X	0.02	objective function coefficient on X
ρ_Y	0.01	objective function coefficient on Y
σ	0.01	weight on objective function control terms
c	2	program cost coefficient
a	2	maximal growth rate at $X = 0$
b	2	maximal growth rate at $Y = 0$
d	1	carrying capacity of Y
α	0.7264	rich to poor coefficient
β	0.5	flight coefficient
γ	0.45	assimilation coefficient
k	1	social integration coefficient
e	1	exponent in the social advancement term

procedure:

```
ocEPn=calcep(n,rand(4,10),opt); c=isadmissible(n,ocEPn,opt);ocEPn(~c)=[];
c=isnegativestate(n,ocEPn);ocEPn(c==1)=[]; ocEPn=uniqueoc(ocEPn,opt); ocEPn{:}
```

which indeed delivers us the above solution:

```
ans =

  dynprimitive object:

      Coordinates:
             0.72232
             1.2002
             0.014808
             0.0037542

      Arc identifier:  1
      Linearization:   [4x4 double]
```

We can always retrieve this equilibrium by typing `ocEPn{1}`.

Let us also take a look at the stability of this equilibrium. By typing `eig(ocEPn{1})` we get the eigenvalues of the above solution. These are

```
ans =

   -2.9785
   -1.5082
    1.5582
    3.0285
```

This exhibits again a two-dimensional stable manifold because the number of eigenvalues ξ satisfying $Re\xi < 0$ is two. We had the same case in the previous modification.

Now we want to receive a phase diagram. We will start from four different points that will all lead to the equilibrium. We'll take the same kind of points as above which are

$(\hat{X}/2, \hat{Y}/2) = (0.3612, 0.6001)$
$(2\hat{X}, 2\hat{Y}) = (1.4446, 2.4004)$
$(\hat{X}/2, 2\hat{Y}) = (0.3612, 2.4004)$
$(2\hat{X}, \hat{Y}/2) = (1.4446, 0.6001)$

Now we have to start a continuation process from each of the above points towards the equilibrium and vice versa by using:

```
initStruct1=initoccont('extremal',n,'initpoint',[1 2],[0.3612;0.6001],
ocEPn{1},'IntegrationTime',500);
```

The next step is solving the BVP: The toolbox does this buy using the following command:

```
[soln solnn]=occont(n,initStruct1,opt);
```

We go through the same procedure for the other three points, so we type the following commands one after the other:

```
initStruct2=initoccont('extremal',n,'initpoint',[1 2],[1.4446;2.4004],
ocEPn{1},'IntegrationTime',500);

[soln1 solnn1]=occont(n,initStruct2,opt);

initStruct3=initoccont('extremal',n,'initpoint',[1 2],[0.3612;2.4004],
```

```
ocEPn{1},'IntegrationTime',500);

[soln2 solnn2]=occont(n,initStruct3,opt);

initStruct4=initoccont('extremal',n,'initpoint',[1 2],[1.4446,0.6001],
ocEPn{1},'IntegrationTime',500);

[soln3 solnn3]=occont(n,initStruct4,opt);
```

Now we want to have a phase diagram with all four paths. By using the command `hold all` we manage to have different graphs in one diagram. So we type the following commands, one by one:

```
plot(soln.dynVar(1,:),soln.dynVar(2,:))
hold all
plot(soln1.dynVar(1,:),soln1.dynVar(2,:))
hold all
plot(soln2.dynVar(1,:),soln2.dynVar(2,:))
hold all
plot(soln3.dynVar(1,:),soln3.dynVar(2,:))
```

After typing all these commands we get the phase diagram that we intended to create (see Figure 6.7).

Now we will go through the same procedure for $\alpha = 1.5604$, which delivers us $\hat{Y} = 1.1986$, which is very similar to the previous value of $\hat{Y}$ at 1.2002. By typing `b1(:,24)` we get the complete equilibrium including α in the last line.

```
ans =

    0.3888
    1.1986
    0.0274
    0.0037
    1.5604
```

Now we can continue our analysis by using the common procedure from before. First of all we change the parameter α of `NewModel.m` from by using the following command:

```
n=changeparameter(n,'alpha',1.5604)
```

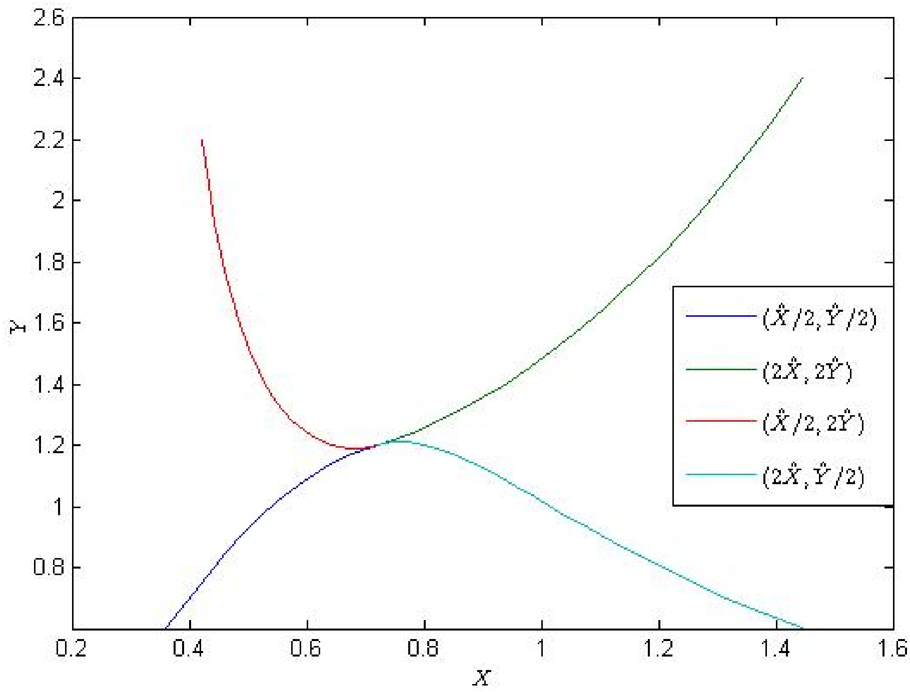

Figure 6.7: Phase diagram for $\alpha = 0.7264$

We find the parameter values of the model in Table 6.3, considering the change that has just been made.

Table 6.3: Modified α Model parameters.

Parameter	Value	Description
r	0.05	discount rate
ρ_X	0.02	objective function coefficient on X
ρ_Y	0.01	objective function coefficient on Y
σ	0.01	weight on objective function control terms
c	2	program cost coefficient
a	2	maximal growth rate at $X = 0$
b	2	maximal growth rate at $Y = 0$
d	1	carrying capacity of Y
α	1.5604	rich to poor coefficient
β	0.5	flight coefficient
γ	0.45	assimilation coefficient
k	1	social integration coefficient
e	1	exponent in the social advancement term

Again we will go through the calculation process in order to receive the equilibrium through the original procedure, too.

```
ocEPn=calcep(n,rand(4,10),opt); c=isadmissible(n,ocEPn,opt);ocEPn(~c)=[];
c=isnegativestate(n,ocEPn);ocEPn(c==1)=[]; ocEPn=uniqueoc(ocEPn,opt);
```

```
ocEPn{:}
```

which indeed delivers us the above solution:

```
ans =

  dynprimitive object:

      Coordinates:
             0.38878
             1.1986
             0.027368
             0.0037401

      Arc identifier:  1
      Linearization:   [4x4 double]
```

We can always retrieve this equilibrium by typing `ocEPn{1}`.
Let us also take a look at the stability of this equilibrium. By typing `eig(ocEPn{1})` we get the eigenvalues of the above solution. These are

```
ans =

   -2.9387
   -1.1401
    1.1901
    2.9887
```

This exhibits again a two-dimensional stable manifold because the number of eigenvalues ξ satisfying $Re\xi < 0$ is two. We had the same case in the previous modifications.

Now we want to receive a phase diagram. We will start from four different points that will all lead to the equilibrium. We'll take the same kind of points as above, which are

$(\hat{X}/2, \hat{Y}/2) = (0.1944, 0.5993)$
$(2\hat{X}, 2\hat{Y}) = (0.7776, 2.3972)$
$(\hat{X}/2, 2\hat{Y}) = (0.1944, 2.3972)$
$(2\hat{X}, \hat{Y}/2) = (0.7776, 0.5993)$

Now we go through our continuation processes from before and solve the corresponding

BVPs. As in the case of the maximum value of $\hat{Y}$ we have to split the first and the third path into two parts in order to get the complete phase diagram. So we type the following commands:

```
initStruct1=initoccont('extremal',n,'initpoint',[1 2],[0.1944;0.5993],
ocEPn{1},'IntegrationTime',500);

[soln solnn]=occont(n,initStruct1,opt);

initStruct1=initoccont('extremal',n,'initpoint',[1 2],[0.1944;0.5993],
solnn,'IntegrationTime',500);

[solna solnna]=occont(n,initStruct1,opt);

initStruct2=initoccont('extremal',n,'initpoint',[1 2],[0.7776;2.3972],
ocEPn{1},'IntegrationTime',500);

[soln1 solnn1]=occont(n,initStruct2,opt);

initStruct3=initoccont('extremal',n,'initpoint',[1 2],[0.1944;2.3972],
ocEPn{1},'IntegrationTime',500);

[soln2 solnn2]=occont(n,initStruct3,opt);

initStruct3=initoccont('extremal',n,'initpoint',[1 2],[0.1944;2.3972],
solnn2,'IntegrationTime',500);

[soln2a solnn2a]=occont(n,initStruct3,opt);

initStruct4=initoccont('extremal',n,'initpoint',[1 2],[0.7776,0.5993],
ocEPn{1},'IntegrationTime',500);

[soln3 solnn3]=occont(n,initStruct4,opt);
```

Now we want to have a phase diagram with all four paths. By using the command `hold all` we manage to have different graphs in one diagram. So we type the following commands, one by one:

```
plot(solna.dynVar(1,:),solna.dynVar(2,:))
hold all
```

```
plot(soln.dynVar(5,:),soln.dynVar(6,:))
hold all
plot(soln1.dynVar(1,:),soln1.dynVar(2,:))
hold all
plot(soln2a.dynVar(1,:),soln2a.dynVar(2,:))
hold all
plot(soln2.dynVar(5,:),soln2.dynVar(6,:))
hold all
plot(soln3.dynVar(1,:),soln3.dynVar(2,:))
```

After typing all these commands we get the phase diagram that we intended to create as displayed in Figure 6.8.

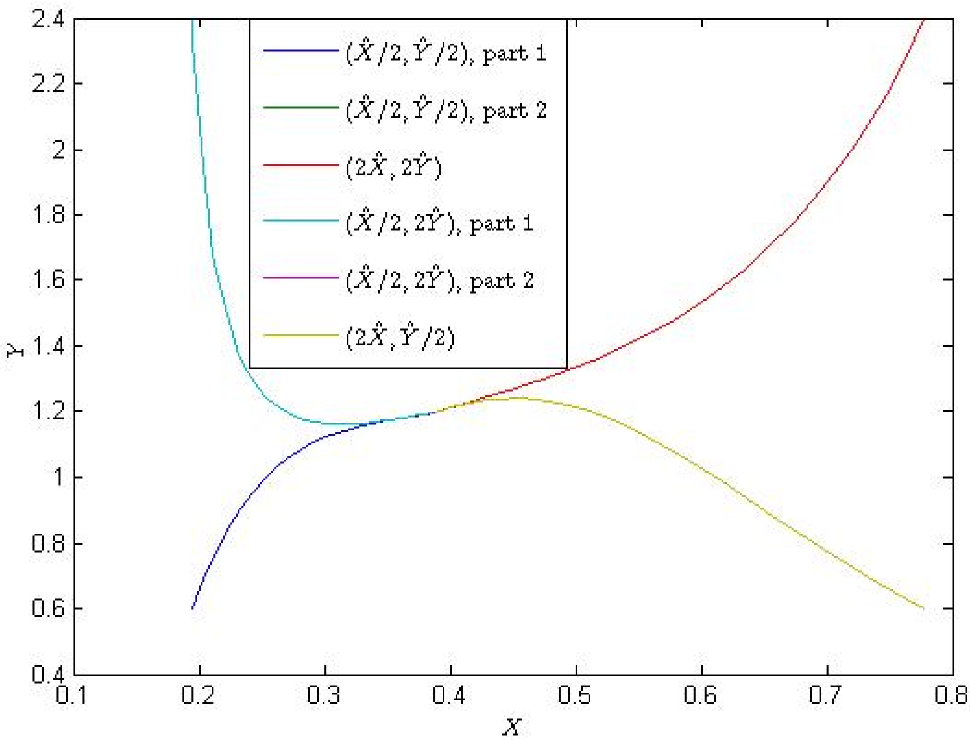

Figure 6.8: Phase diagram for $\alpha = 1.5604$

6.3 Time Paths

Now we will go one step further and compute time paths. We will look at cases, where $\hat{Y}$ has almost the same value, namely $\hat{Y} = 1.2002, 1.1986$ for $\alpha = 0.7264, 1.5604$, respectively. Taking into account the fact that we previously used four different starting points to create the phase potraits, we will now display the solutions as time paths. So we will get four different diagrams for each value of α, corresponding to the different starting points.

$\alpha = 0.7264$

As we have previously mentioned, the equilibrium that we get when setting $\alpha = 0.7264$ is

```
ans =

    0.7223
    1.2002
    0.0148
    0.0038
```

Now we will take a look at two different diagrams, considering the following starting points:

$(\hat{X}/2, \hat{Y}/2)$
$(2\hat{X}, 2\hat{Y})$
$(\hat{X}/2, 2\hat{Y})$
$(2\hat{X}, \hat{Y}/2)$

We will have t on the x-axis and $\hat{X}, \hat{Y}, \hat{u}$ on the y-axis. We will again use the command `hold all` in order to make sure that we have all three graphs in one diagram, starting at $t = 0$. For $(\hat{X}/2, \hat{Y}/2)$ we have the following commands:

```
plot(soln.t*soln.timeintervals,soln.dynVar(1,:))
hold all
plot(soln.t*soln.timeintervals,soln.dynVar(2,:))
hold all
plot(soln.t.*soln.timeintervals,control(n,soln))
```

We go through the same procedure for the remaining three points and receive the following diagrams, displayed in Figures 6.9-6.12.

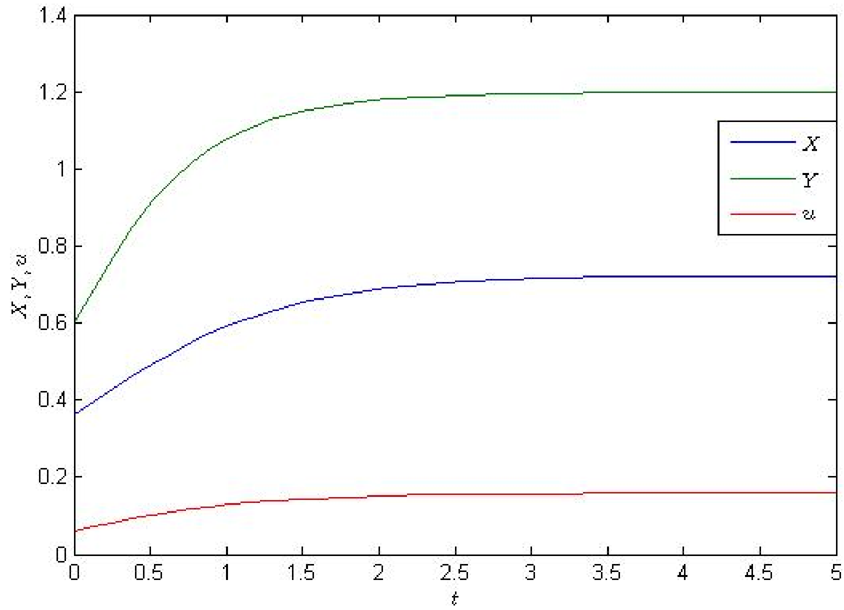

Figure 6.9: $\alpha = 0.7264$, starting point: $(\hat{X}/2, \hat{Y}/2)$; $X(t)$ and $Y(t)$ are increasing. $u(t)$ is also slightly increasing. $Y(t)$ is always above $X(t)$, $u(t)$ is a lot lower.

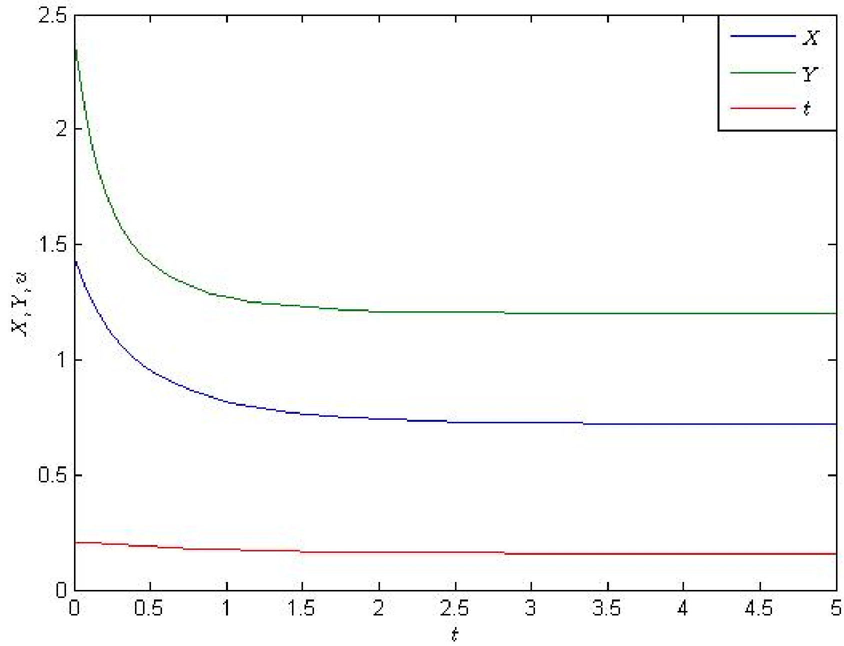

Figure 6.10: $\alpha = 0.7264$, starting point: $(2\hat{X}, 2\hat{Y})$; $X(t)$ and $Y(t)$ are decreasing, $u(t)$ is also slightly decreasing. $Y(t)$ is always above $X(t)$, they develop proportionally to each other. $u(t)$ is a lot lower.

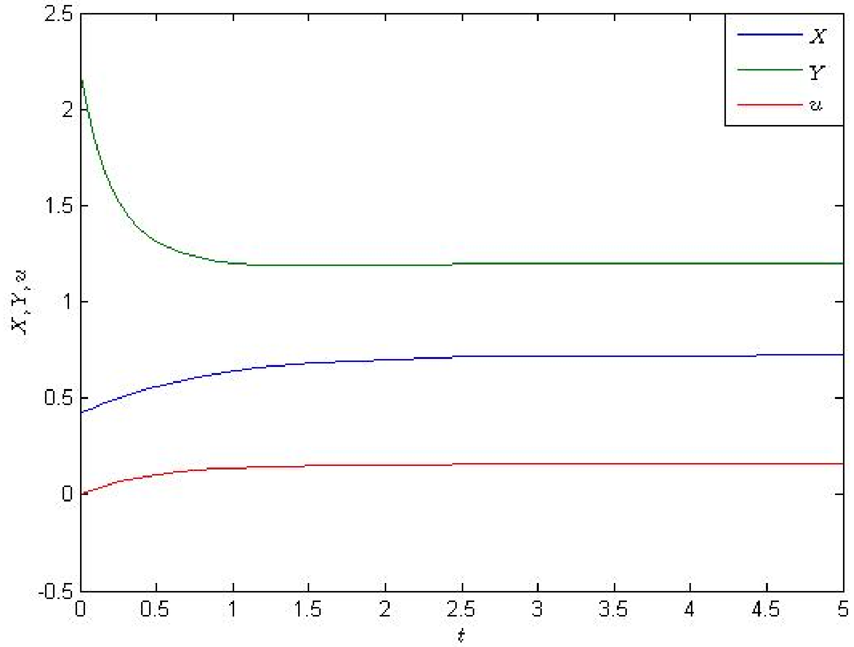

Figure 6.11: $\alpha = 0.7264$, starting point: $(\hat{X}/2, 2\hat{Y})$; $Y(t)$ is decreasing but stays above $X(t)$ and $u(t)$, which are increasing.

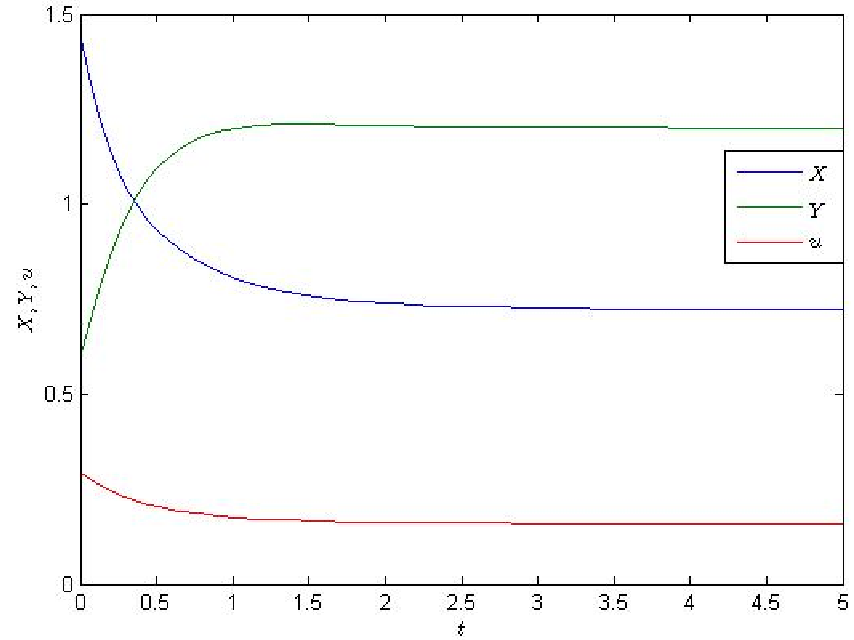

Figure 6.12: $\alpha = 0.7264$, starting point: $(2\hat{X}, \hat{Y}/2)$; $Y(t)$ is increasing, $X(t)$ and $u(t)$ are decreasing. In the beginning, $X(t)$ is above $Y(t)$, but then they switch so that $Y(t)$ is above $X(t)$. $u(t)$ is a lot lower.

$\alpha = 1.5604$

The equilibrium that we get when setting $\alpha = 1.5604$ is

```
ans =
     0.3888
     1.1986
     0.0274
     0.0037
```

We will only consider the starting points $(2\hat{X}, 2\hat{Y})$ and $(2\hat{X}, \hat{Y}/2)$ as the command `occont` does not compute sufficient solution paths in the other two cases. We will go through the same procedure as before in order to receive the time paths. We will have t on the x-axis and $\hat{X}, \hat{Y}, \hat{u}$ on the y-axis. Repeating the same procedure as in the previous section we get the following two diagrams in Figures 6.13 and 6.14.

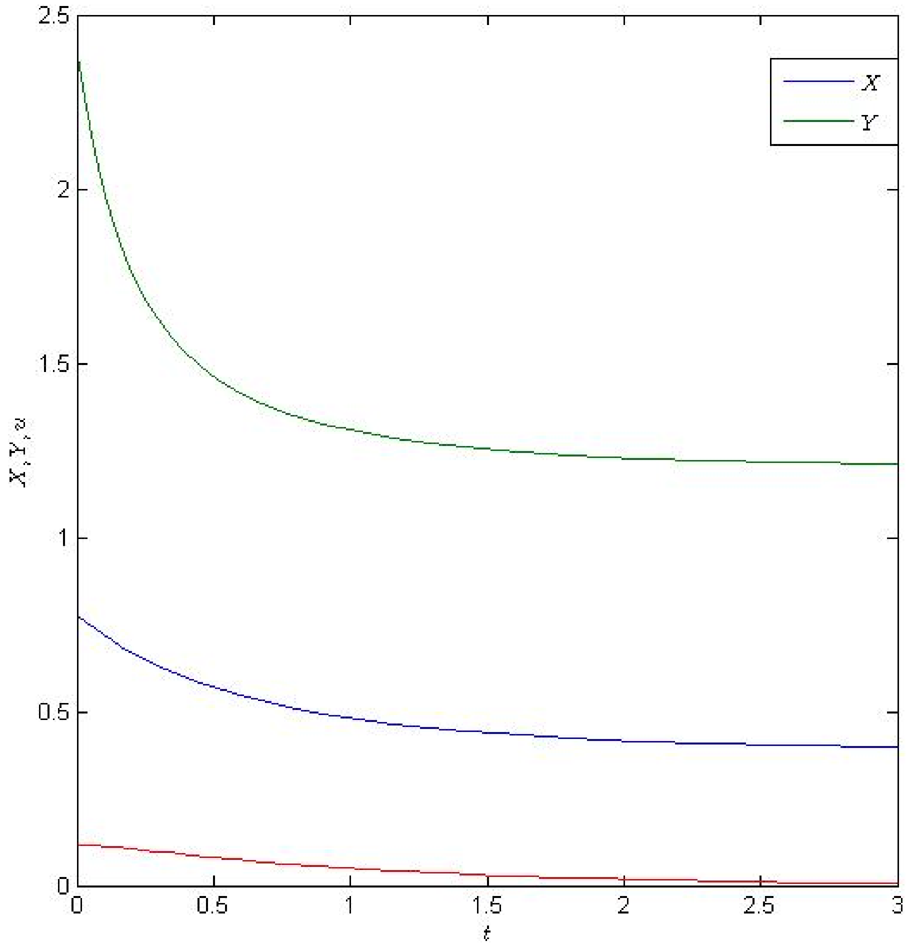

Figure 6.13: $\alpha = 1.5604$, starting point: $(2\hat{X}, 2\hat{Y})$; This figure is similar to Figure 6.10. $Y(t)$ is decreasing, $X(t)$ and $u(t)$ are slightly decreasing. $Y(t)$ is a lot higher than $X(t)$ and $u(t)$.

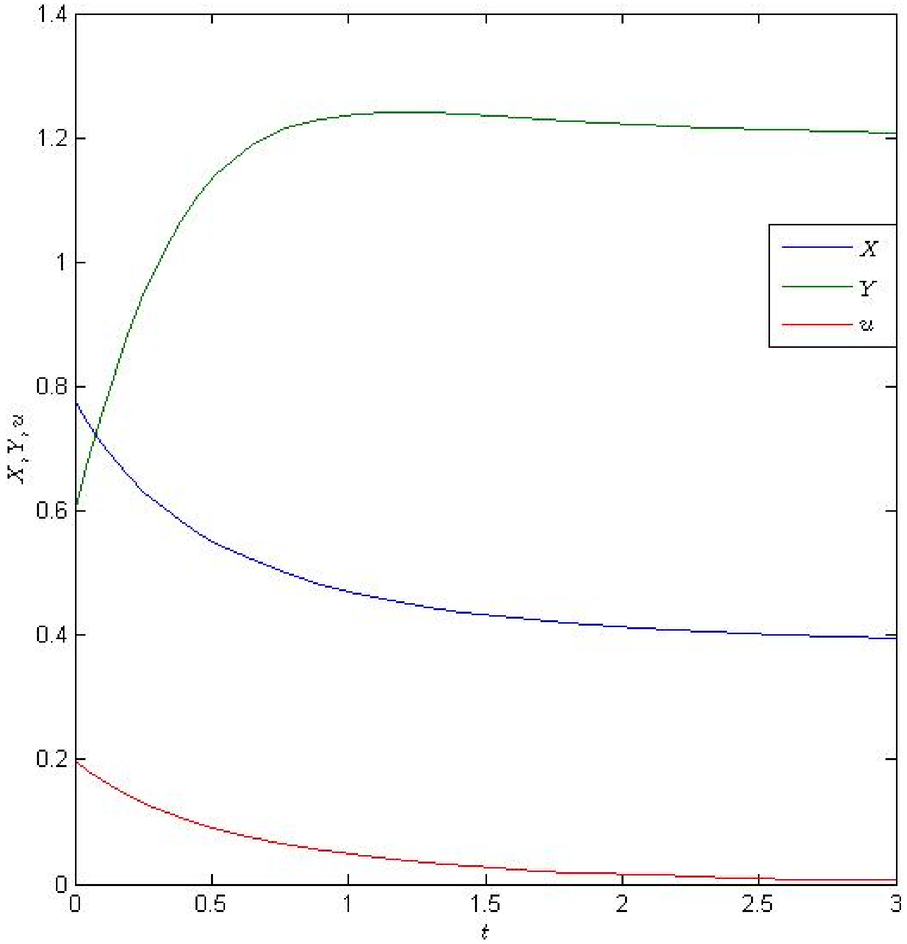

Figure 6.14: $\alpha = 1.5604$, starting point: $(2\hat{X}, \hat{Y}/2)$; This figure is similar to Figure 6.12. In the beginning, X is higher than Y, but then they switch and Y becomes a lot bigger than X because Y is increasing whereas X and u are decreasing.

CHAPTER 7

The β Model

In the previous chapter we introduced a parameter α for modelling the case in which rich people living in the middle-class area become "poor" in the course of time. So we subtracted the term $\alpha X(t)$ from the $\dot{X}$ equation and added this term to the $\dot{Y}$ equation. Now we go back to the original model but implement a different modification. This time there is only a change in the $\dot{X}$ equation. We substitute the variable β by β_1 and subtract another term $\beta_2 Y$. So the new model which we will call `BetaModel.m` looks as follows:

$$\max_{u(\cdot)\geq 0} \int_0^\infty e^{-rt} \left[\rho_X X(t) + \rho_Y Y(t) + \sigma(u(t) - cu(t)^2)\right] dt \tag{7.1}$$

subject to the dynamic state equations

$$\dot{X}(t) = aX(t)(1 - X(t)) - \beta_1 u(t) - \beta_2 Y + \gamma Y(t) \left(\frac{kX(t)}{kX(t) + Y(t)}\right)^e, \tag{7.2}$$

$$\dot{Y}(t) = bY(t)(d - Y(t)) - \gamma Y(t) \left(\frac{kX(t)}{kX(t) + Y(t)}\right)^e + u(t). \tag{7.3}$$

Here we will have another restriction, namely that

$$\beta_1 + \beta_2 = 0.5.$$

In order to make sure that this restriction is fulfilled at all times we make another change in the model, namely change β_1 back to β and put $(0.5 - \beta)$ instead of β_2. This makes our further work a lot easier.

This modification takes into account the debate, whether middle-class flight is induced by the flow of immigrants or the stock of poor people in the neighbourhood. Varying β between 0.5 and 0 means putting different emphasis on these mechanisms.

As we are dealing with a new model we have to go through the initialisation process again. And then we check the case $\beta = 0.5$ to make sure that we get the same results as in the original case. Now we can concentrate on the cases where we have different values for β and see how the optimal solution changes. We will also examine the other equilibria.

7.1 Continuing the Equilibria with Matcont

The next step is to change the parameter β in order to see how this influences the values of X and Y and especially the optimal solution $(\hat{X}, \hat{Y}, \hat{\lambda}_1, \hat{\lambda}_2)$. Before having a look at specific values of β and analysing those in more detail we want to check what happens with $\hat{X}$ and $\hat{Y}$ if β becomes bigger and bigger. So we want to continue the original equilibria. We will follow the same procedure as in the previous chapter by using the toolbox Matcont.

Continuing the equilibria happens with the following command:
`[c1,c2,c3]=contep(o,ocEPo{1},'beta',opt)`
where `o` stands for `BetaModel.m`. Before we do this we need to check if β is becoming bigger or smaller. This happens with the command `opt.MATCONT.Backward=1` or `opt.MATCONT.Backward=[]`. `c1` stands for the equibrium which is continued. After doing this we receive 300 equilibra.

Now that we have continued the original equilibrium we can have a look at the bifurcation diagrams. We first take a look at the two-dimensional diagrams before putting them together in a three-dimensional version.

If we want to see how $\hat{X}$ develops if we increase β we need to type the command `plot(c1(end,:),c1(1,:))`. We find the diagram in Figure 7.1 and also another diagram in Figure 7.2 where β lies between the reasonable values 0 and 0.5. We will have β on the x-axis and $\hat{X}$ on the y-axis. So in the command that we used above `c1(end,:)` stands for the parameter β which is being increased and `c1(1,:)` stands for $\hat{X}$.

As a next step we will have a look at the diagram where $\hat{Y}$ is continued. We use a very similar command: `plot(c1(end,:),c1(2,:))` as the only difference we have is exchanging $\hat{X}$ by $\hat{Y}$ (Figure 7.3). We will also take a look at the diagram where β is limited between 0 and 0.5 (Figure 7.4).

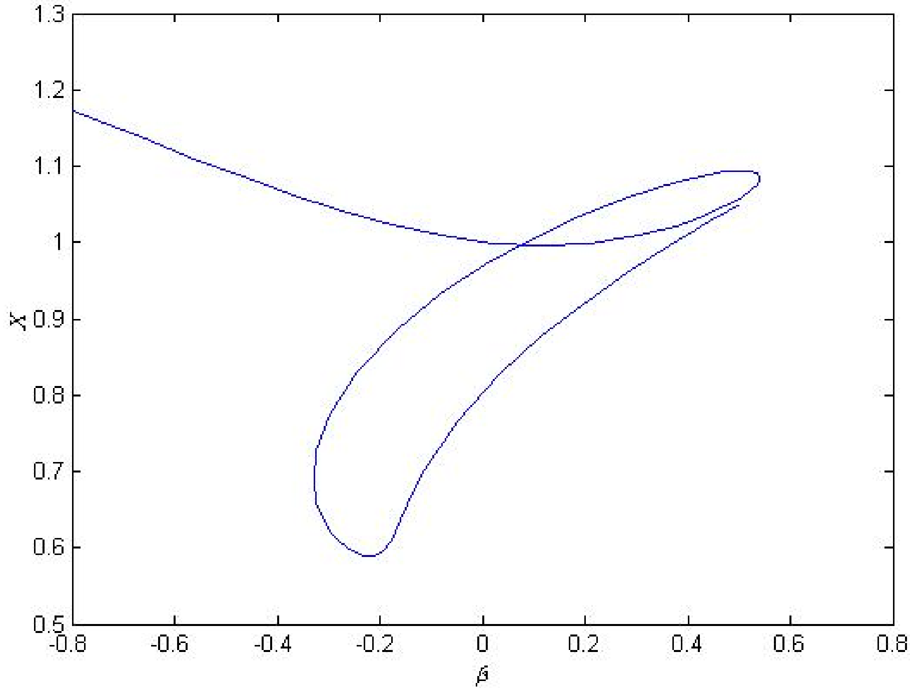

Figure 7.1: Continuing $\hat{X}$ by increasing β

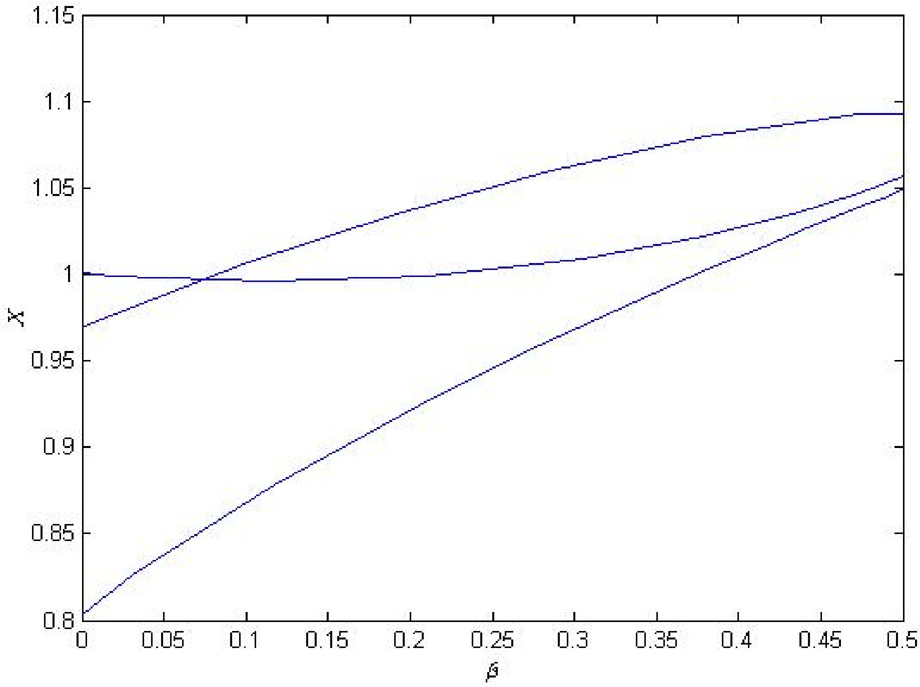

Figure 7.2: Continuing $\hat{X}$ by increasing β

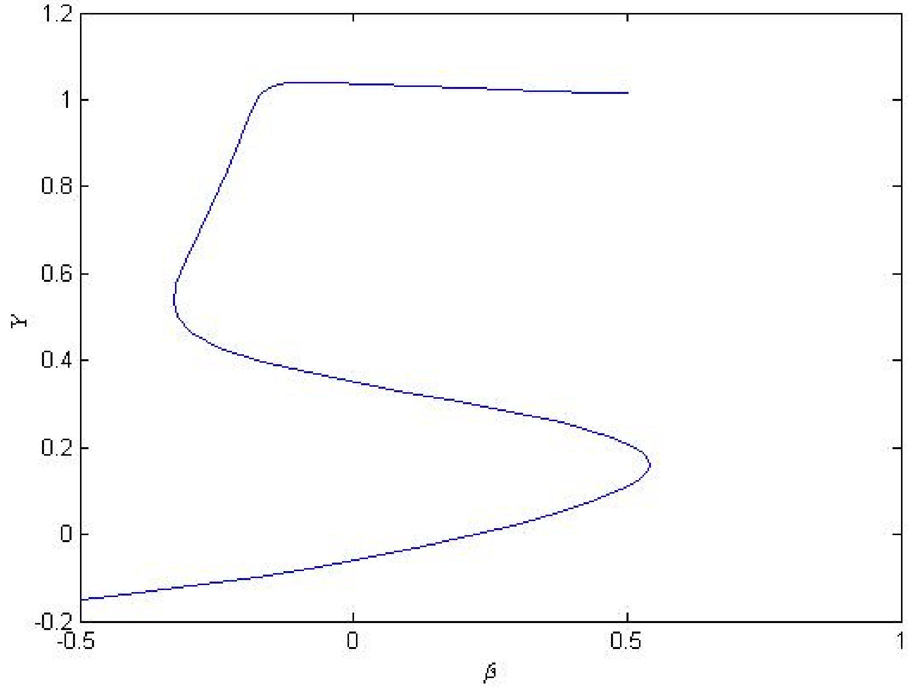

Figure 7.3: Continuing $\hat{Y}$ by increasing β

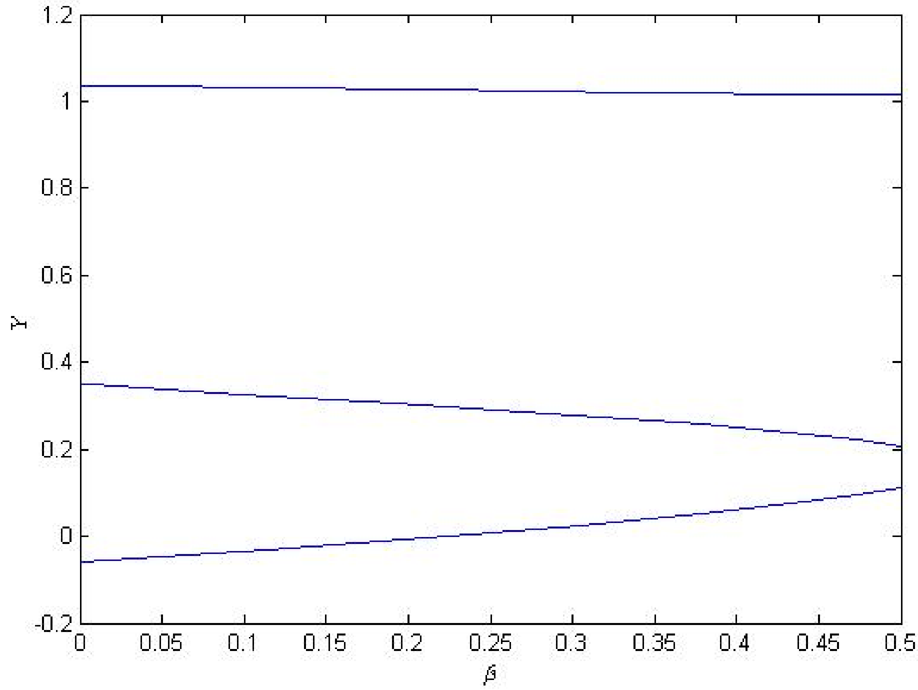

Figure 7.4: Continuing $\hat{Y}$ by increasing β

In addition to the diagrams we have created so far we will now have a look at the three-dimesional digram where we see the development of $\hat{X}$ and $\hat{Y}$ at the same time. We have to use the command `plot3(c1(1,:),c1(2,:),c1(end,:))`. So we have $\hat{X}$ on the x-axis, $\hat{Y}$ on the y-axis and β on the z-axis (Figure 7.5). And as a last diagram we will again look at the case where β lies between 0 and 0.5 (Figure 7.6).

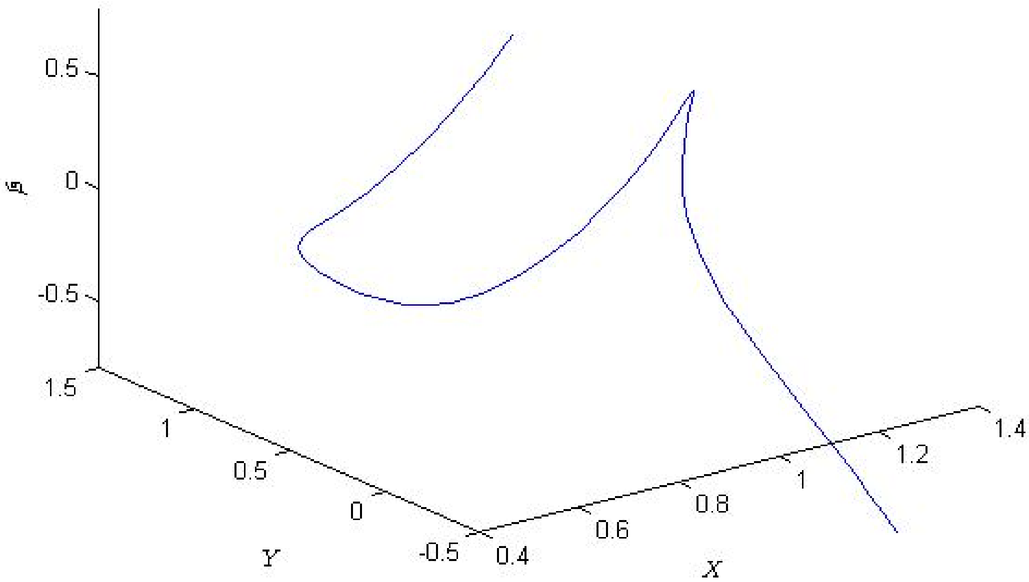

Figure 7.5: Continuing $\hat{X}$ and $\hat{Y}$ by increasing β

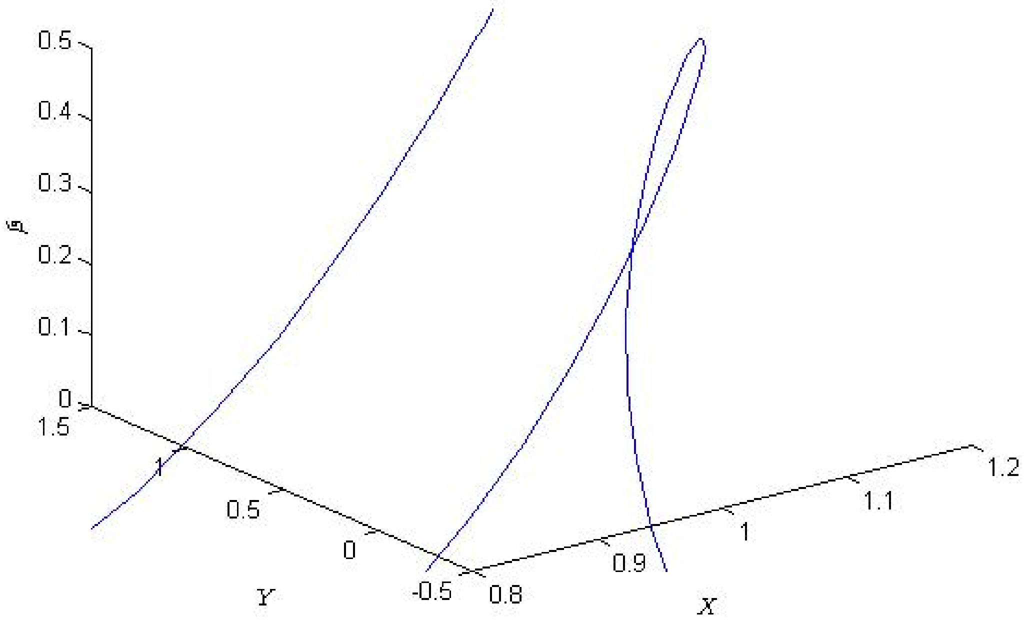

Figure 7.6: Continuing $\hat{X}$ and $\hat{Y}$ by increasing β

As in the previous chapter we will now also create a bifurcation diagrams where $\hat{X}, \hat{Y}$ and the control $\hat{u}$ are displayed at the same time, all depending on β. First of all we need to get hold of the control $\hat{u}$. We do this by re-using the function `controlalpha` from before:

```
out = controlalpha(n,c1,1)
```

This function delivers us the control values of the whole path. So now we have all the inputs that we need for the joint bifurcation diagram. But first we will have a look at the phase potrait that shows us the relation of the control $\hat{u}$ and the parameter β (Figure 7.7).

```
plot(c1(end,:),out)
```

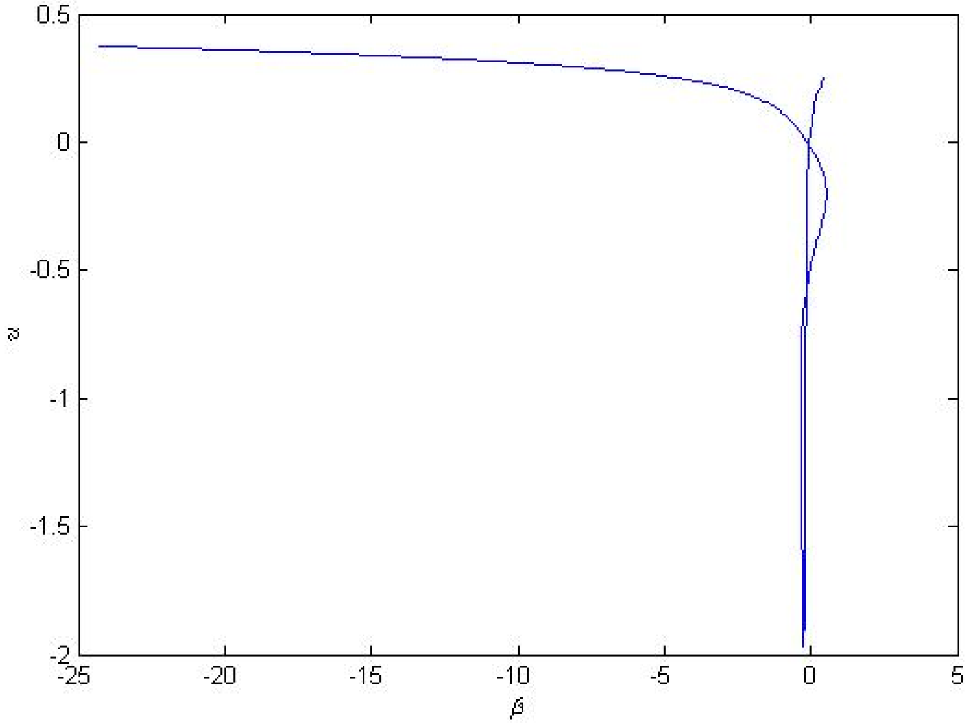

Figure 7.7: Continuing $\hat{u}$ by increasing β

Since we cannot have a four-dimensional diagram we will go back to the two-dimensional version and have all three graphs for $\hat{X}, \hat{Y}$ and $\hat{u}$ in one diagram by using the command `hold all` (Figure 7.8).

```
plot(c1(end,:),out)
hold all
plot(c1(end,:),c1(1,:))
hold all
plot(c1(end,:),c1(2,:))
hold all
plot(c1(end,:),out)
```

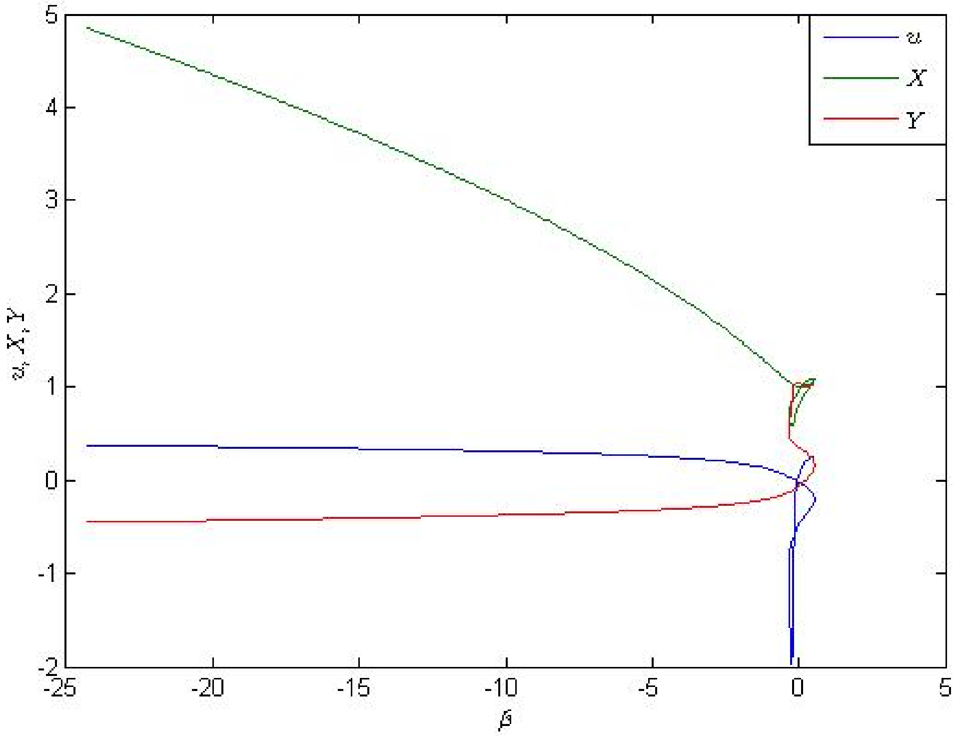

Figure 7.8: Continuing $\hat{X}, \hat{Y}$ and $\hat{u}$ by increasing β

We see some interesting findings that will be helpful for our further analysis of specific values of β. To be more precise we take a look at the actual values of $\hat{X}, \hat{Y}$ and $\hat{u}$ by typing `c1` and `out`. There we see that $\hat{Y}$ becomes negative starting from index 52. $\hat{X}$ is always positive but $\hat{u}$ is negative in the interval between index 13 and 55. This means that we should only consider the first 12 indices, which fit our original restriction, namely that β lies between 0 and 0.5. If we only consider the admissible area, the two diagrams in Figures 7.7. and 7.8 should look as the ones in Figures 7.9 and 7.10, respectively.

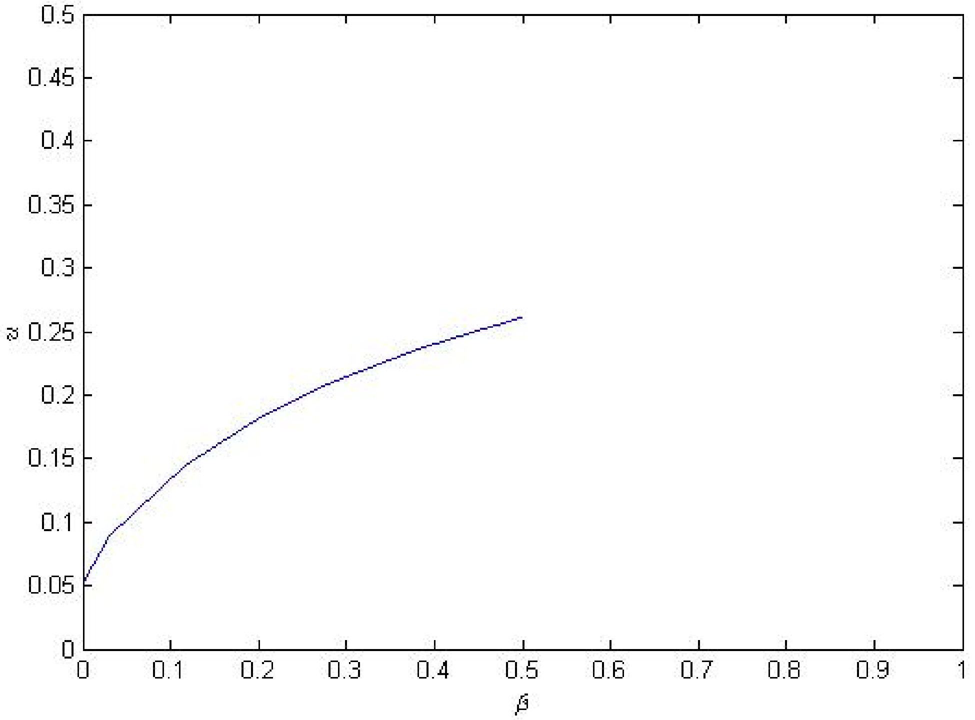

Figure 7.9: Continuing $\hat{u}$ by increasing β

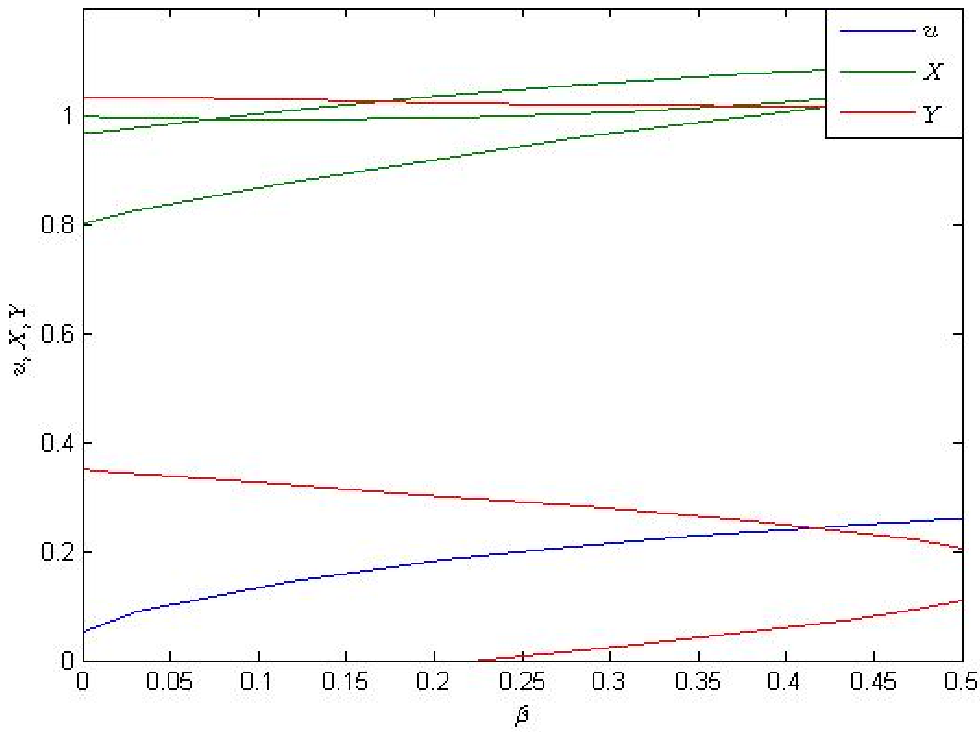

Figure 7.10: Continuing $\hat{X}, \hat{Y}$ and $\hat{u}$ by increasing β

The Case $\beta_1 = 0.49, \beta_2 = 0.01$

In the following sections we will actually work with the original parameters β_1 and β_2 that we introduced first as we are controlling the ristriction $\beta_1 + \beta_2 = 0.5$ manually. As we are not sure how the model will react if we introduce β_2, we will start from a very small value, i.e. $\beta_2 = 0.01$. As mentioned above we have the restriction that β_1 and β_2 together should be equal to 0.5. Therefore in this case β_1 equals 0.49.

So now we will continue with our usual analysis. First of all we need to calculate the equilibria. Again we will use the command `calcep`.

```
ocEPo=calcep(o,rand(4,10),opt); d=isadmissible(o,ocEPo,opt);ocEPo(~d)=[];
d=isnegativestate(o,ocEPo);ocEPo(d==1)=[]; ocEPo=uniqueoc(ocEPo,opt);
ocEPo{:}
```

We retrieve four equilibra:

```
ans =

  dynprimitive object:

      Coordinates:
             0.1084
             1.1771
             -0.0098349
             0.0036453

      Arc identifier:  1
      Linearization:   [4x4 double]



ans =

  dynprimitive object:

      Coordinates:
             1.0448
             1.0144
             0.0091797
             0.0049342
```

```
      Arc identifier:  1
      Linearization:   [4x4 double]


ans =

  dynprimitive object:

      Coordinates:
             0.49103
             1.4116
             -0.083257
             0.0022374

      Arc identifier:  1
      Linearization:   [4x4 double]


ans =

  dynprimitive object:

      Coordinates:
             1
             -3.7082e-010
             0.0097561
             -0.0095285

      Arc identifier:  2
      Linearization:   [4x4 double]
```

If we have a look at the second equilibrium we easily realise that this equilibrium is quite similar to the optimal solution of the original model, which was $\hat{X} = 1.0485$, $\hat{Y} = 1.0141$, $\hat{\lambda}_1 = 0.009113$ and $\hat{\lambda}_2 = 0.0049762$. If we take a look at the corresponding equilibrium in our current model we see that $\hat{X}$ and $\hat{\lambda}_2$ have decreased whereas $\hat{Y}$ and $\hat{\lambda}_1$ have increased. But as we already mentioned when we analysed the α model, it is only an assumption that `ocEPo{2}` is the optimal solution, so we still need to prove that this statement is actually correct. As before we will use two different ways of proving it. To analyse the first and the second equilibrium, we can start a continuation process from the state of the second into the first equilibrium and vice versa by using:

```
initStruct2=initoccont('extremal',o,'initpoint',[1 2],ocEPo{1}.dynVar(1:2,1),
ocEPo{2},'IntegrationTime',500);
```

In the underlying case, the above initialized continuation is successful, i.e., the first equilibrium can be excluded.

Now we will proceed with calculating the Hamiltonian function. But before doing that we have to solve the BVP. The toolbox solves by

```
[solo solno]=occont(o,initStruct2,opt);
```

To retrieve a result (already stored in `o`) one can use `ocEPo=equilibrium(o)` for retrieving the elements of the fields of the equilibrium and `ocExn=extremalsol(o)` for retrieving the elements of the field corresponding to the stable path. They are stored in `ocResults` among other calculated elements. Note that one can check all the stored calculations made before by calling `o.ocResults`.

Now that we have calculated the BVP we can go back to the Hamiltonian. The next task is to compare the Hamiltonian of the first equilibrium with the Hamiltonian of the BVP. For the first Hamiltonian we use the command `hamiltonian(o,ocEPo{1})` and get the following result

```
ans =

    0.0143    0.0143
```

For the second Hamiltonian we use the command `hamiltonian(o,solo)` and get the following results

```
ans =

  Columns 1 through 10

    0.0318    0.0318    0.0318    0.0319    0.0320    0.0320    0.0321
0.0321    0.0322    0.0322

  Columns 11 through 20
```

```
    0.0322    0.0322    0.0322    0.0323    0.0323    0.0323    0.0323
0.0323    0.0323    0.0323

  Columns 21 through 30

    0.0323    0.0323    0.0323    0.0323    0.0323    0.0323    0.0323
0.0323    0.0323    0.0323

  Columns 31 through 39

    0.0323    0.0323    0.0323    0.0323    0.0323    0.0323    0.0323
0.0323    0.0323
```

As we see, the first Hamiltonian is clearly smaller than the second one. The rule is that if the command `hamiltonian(o,ocEPo{1})` delivers a smaller result than the command `hamiltonian(o,solo)` we know that the first equilibrium is not an optimal solution. Thus we stick with the assumption that the second equilibrium compared to the first equilibrium is the optimal solution.

Now we go through the same procedure using the third and fourth equlibrium. So we re-use the commands from above and eventually get the same results, namely that the second equilibrium is indeed the optimal solution.

The next step is having a look at the phase diagram of the optimal solution (Figure 7.11). We will have X on the x-axis and Y on the y-axis and use the same starting points as in the precious chapter, which are

$(\hat{X}/2, \hat{Y}/2) = (0.5224, 0.5072)$
$(2\hat{X}, 2\hat{Y}) = (2.0896, 2.0288)$
$(\hat{X}/2, 2\hat{Y}) = (0.5224, 2.0288)$
$(2\hat{X}, \hat{Y}/2) = (2.0896, 0.5072)$.

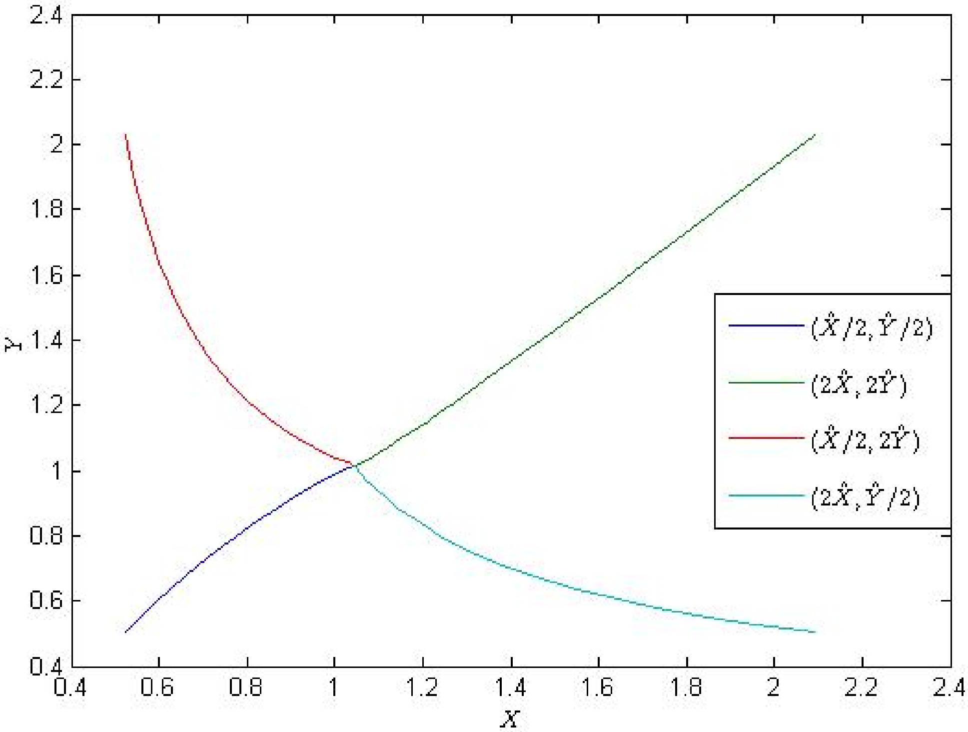

Figure 7.11: $\beta_1 = 0.49, \beta_2 = 0.01$, X in relation to Y

The Case $\beta_1 = 0.25, \beta_2 = 0.25$

As we have seen, there were no big changes for the previous case. Now we will have a look at the case where β_1 equals β_2. Thus both have to take the value 0.25. So let us start right away with calculating the equilibria:

```
ocEPo=calcep(o,rand(4,10),opt); d=isadmissible(o,ocEPo,opt);ocEPo(~d)=[];
d=isnegativestate(o,ocEPo);ocEPo(d==1)=[]; ocEPo=uniqueoc(ocEPo,opt); ocEPo{:}
```

We get the following results:

```
ans =

  dynprimitive object:

      Coordinates:
             0.94577
             1.0243
             0.011423
             0.0037001
```

```
      Arc identifier:  1
      Linearization:   [4x4 double]


ans =

  dynprimitive object:

      Coordinates:
             0.51229
             1.3967
             -0.12397
             0.010077

      Arc identifier:  1
      Linearization:   [4x4 double]


ans =

  dynprimitive object:

      Coordinates:
             1
             -1.4467e-008
             0.0097561
             -0.0079671

      Arc identifier:  2
      Linearization:   [4x4 double]
```

Again it is easy to see that the first equibrium is very similar to the optimal solution of the previous model which was $\hat{X} = 1.0448$, $\hat{Y} = 1.0144$, $\hat{\lambda}_1 = 0.0091797$ and $\hat{\lambda}_2 = 0.0049342$. We have the same development as in the previous model because $\hat{X}$ and $\hat{\lambda}_2$ have decreased whereas $\hat{Y}$ and $\hat{\lambda}_1$ have increased.

Now we will proceed with proving that `ocEPo{1}` indeed is the optimal solution. As before we will use two different ways of proving it. To analyse the first and the second equilibrium, we can start a continuation process from the state of the second into the first equilibrium and vice versa by using:

```
initStruct2=initoccont('extremal',o,'initpoint',[1 2],ocEPo{2}.dynVar(1:2,1),
ocEPo{1},'IntegrationTime',500);
```

In the underlying case, the above initialized continuation is successful, i.e., the second equilibrium can be excluded.

Now we will proceed with calculating the Hamiltonian function. But before doing that we have to solve the BVP. The toolbox solves by

```
[solo solno]=occont(o,initStruct2,opt);
```

For the first Hamiltonian we use the command `hamiltonian(o,ocEPo{2})` and get the following result

```
ans =

  0.0044    0.0044
```

For the second Hamiltonian we use the command `hamiltonian(o,solo)` and get the following results

```
ans =

  Columns 1 through 10

    0.0301    0.0301    0.0301    0.0301    0.0301    0.0302    0.0302
0.0302    0.0302    0.0303

  Columns 11 through 20

    0.0303    0.0303    0.0303    0.0303    0.0304    0.0304    0.0304
0.0304    0.0304    0.0304

  Columns 21 through 30

    0.0304    0.0304    0.0304    0.0304    0.0304    0.0304    0.0304
0.0304    0.0304    0.0304
```

```
  Columns 31 through 35

    0.0304    0.0304    0.0304    0.0304    0.0304
```

As we see, the first Hamiltonian is clearly smaller than the second one. The rule is that if the command `hamiltonian(o,ocEPo{2})` delivers a smaller result than the command `hamiltonian(o,solo)` we know that the second equilibrium is not an optimal solution. Thus we know for sure that the first equilibrium in comparison with the second equilibrium is the optimal solution.

Now we go through the same procedure using the third equilibrium. So we re-use the commands from above and eventually get the same results, namely that the first equilibrium is indeed the optimal solution.

The next step is having a look at the phase diagram of the optimal solution (Figure 7.12). We will have X on the x-axis and Y on the y-axis. This time we have the following starting points:

$(\hat{X}/2, \hat{Y}/2) = (0.4729, 0.5121)$
$(2\hat{X}, 2\hat{Y}) = (1.8915, 2.0486)$
$(\hat{X}/2, 2\hat{Y}) = (0.4729, 2.0486)$
$(2\hat{X}, \hat{Y}/2) = (1.8915, 0.5121)$.

The Case $\beta_1 = 0, \beta_2 = 0.5$

If we compare the current model with the original model, we would have to choose $\beta_1 = 0.5$ and $\beta_2 = 0$. Now we will have a look at the opposite case where $\beta_1 = 0$ and $\beta_2 = 0.5$. This means that the term $\beta_1 u$ disappears completely.

Again we start by calculating the equilibra:

```
ocEPo=calcep(o,rand(4,10),opt); d=isadmissible(o,ocEPo,opt);ocEPo(~d)=[];
d=isnegativestate(o,ocEPo);ocEPo(d==1)=[]; ocEPo=uniqueoc(ocEPo,opt); ocEPo{:}
```

We get the following results:

```
ans =
```

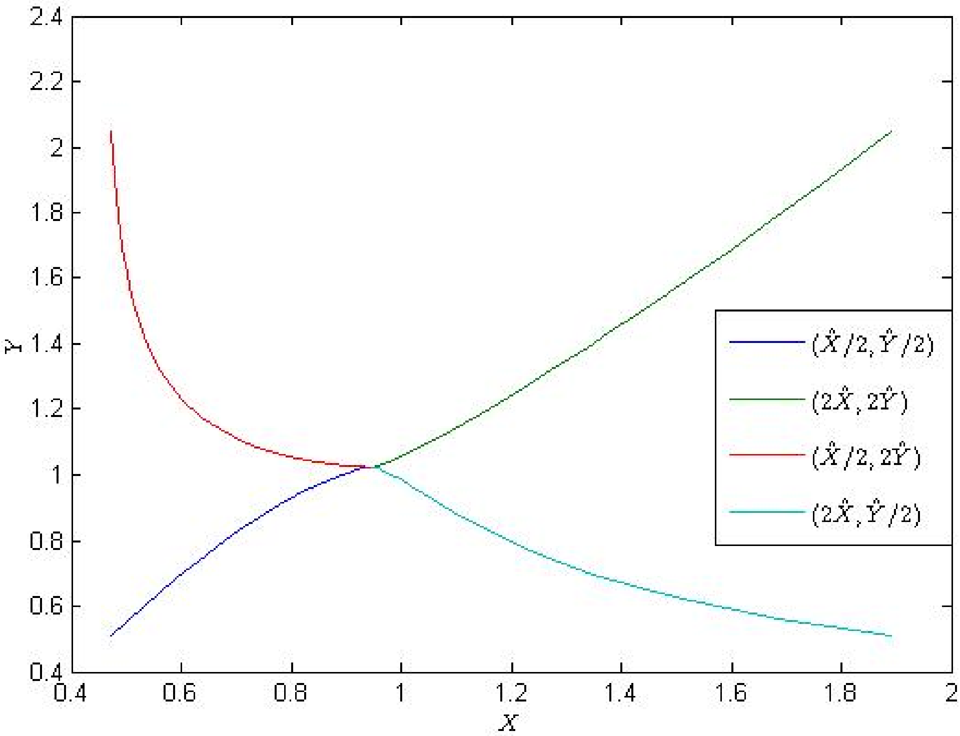

Figure 7.12: $\beta_1 = 0.25, \beta_2 = 0.25$, X in relation to Y

```
  dynprimitive object:

      Coordinates:
             0.53132
             1.3384
             -0.22382
             0.033076

      Arc identifier:  1
      Linearization:   [4x4 double]


ans =

  dynprimitive object:

      Coordinates:
             0.8046
```

```
            1.0366
            0.017616
            0.0011857

    Arc identifier:  1
    Linearization:   [4x4 double]
```

This time the second equilibrium is fairly similar to the optimal solution of the previous model which was $\hat{X} = 0.94577$, $\hat{Y} = 1.0243$, $\hat{\lambda}_1 = 0.011423$ and $\hat{\lambda}_2 = 0.0037001$. We have the same development as in the previous model because $\hat{X}$ and $\hat{\lambda}_2$ have decreased whereas $\hat{Y}$ and $\hat{\lambda}_1$ have increased.

Now we will proceed with proving that `ocEPo{2}` indeed is the optimal solution. As before we will use two different ways of proving it. To analyse the first and the second equilibrium, we can start a continuation process from the state of the first into the second equilibrium and vice versa by using:

```
initStruct2=initoccont('extremal',o,'initpoint',[1 2],ocEPo{1}.dynVar(1:2,1),
ocEPo{2},'IntegrationTime',500);
```

In the underlying case, the above initialised continuation is successful, i.e., the second equilibrium can be excluded.

Now we will proceed with calculating the Hamiltonian function. But before doing that we have to solve the BVP. The toolbox solves by

```
[solo solno]=occont(o,initStruct2,opt);
```

For the first Hamiltonian we use the command `hamiltonian(o,ocEPo{1})` and get the following result

```
ans =

    0.0116    0.0116
```

For the second Hamiltonian we use the command `hamiltonian(o,solo)` and get the following results

```
ans =
```

```
  Columns 1 through 10

    0.0273    0.0273    0.0273    0.0273    0.0274    0.0274    0.0274
0.0274    0.0275    0.0275

  Columns 11 through 20

    0.0275    0.0275    0.0276    0.0276    0.0276    0.0276    0.0276
0.0277    0.0277    0.0277

  Columns 21 through 30

    0.0277    0.0277    0.0277    0.0277    0.0277    0.0277    0.0277
0.0277    0.0277    0.0277

  Columns 31 through 40

    0.0277    0.0277    0.0277    0.0277    0.0277    0.0277    0.0277
0.0277    0.0277    0.0277

  Columns 41 through 45

    0.0277    0.0277    0.0277    0.0277    0.0277
```

As we see, the first Hamiltonian is clearly smaller than the second one. Thus we know for sure that the second equilibrium in comparison with the first equilibrium is the optimal solution.

The next step is having a look at the phase diagram of the optimal solution (Figure 7.13). We will have X on the x-axis and Y on the y-axis and use the following starting points:

$(\hat{X}/2, \hat{Y}/2) = (0.4023, 0.5183)$
$(2\hat{X}, 2\hat{Y}) = (1.6092, 2.0732)$
$(\hat{X}/2, 2\hat{Y}) = (0.4023, 2.0732)$
$(2\hat{X}, \hat{Y}/2) = (1.6092, 0.5183)$.

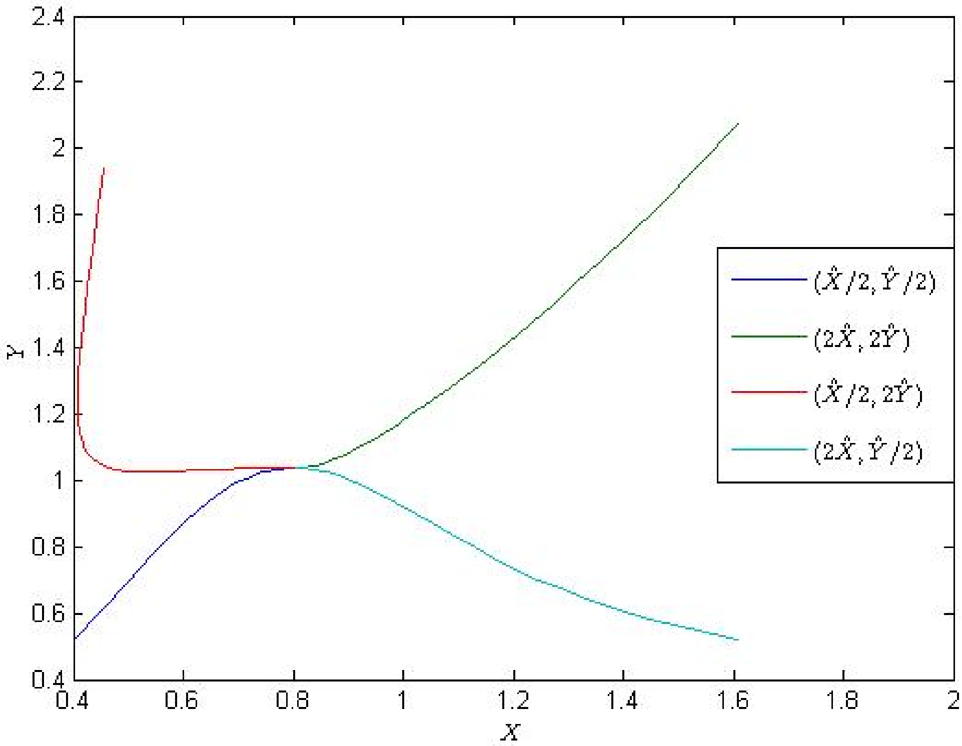

Figure 7.13: $\beta_1 = 0, \beta_2 = 0.5$, X in relation to Y

7.2 DNSS Curve

As mentioned in the Introduction, multiple equilibria are related to the existence of so-called Skiba or DNSS points. As we have seen in the analysis above, there are no multiple optimal solutions in the β Model. However, we can change the parameters in order to see if we can modify the existing model in such a way so that we find a DNSS point. We will do so by changing the parameter ρ_X. The reason why we chose ρ_X is that X seems to be of big importance for the model. The optimal solution always has a big value of $\hat{X}$. So we take away the emphasis of the variable X in order to see if we have multiple optimal solutions afterwards.

ρ_X has already a rather small value of 0.02 in base case. After trying different values by making β smaller and smaller, I found out that there are some interesting changes for $\rho_X = 0.0005$. In the following I will describe how it is possible to find a DNSS point and finally get a diagram with a DNSS curve.

First of all we set $\rho_X = 0.0005$ by the known command

```
o=changeparameter(o,'rhox',0.0005)
```

So now we have the parameters for the current model as given in Table 7.1.

Table 7.1: β Model parameters.

Parameter	Value	Description
r	0.05	discount rate
ρ_X	0.0005	objective function coefficient on X
ρ_Y	0.01	objective function coefficient on Y
σ	0.01	weight on objective function control terms
c	2	program cost coefficient
a	2	maximal growth rate at $X = 0$
b	2	maximal growth rate at $Y = 0$
d	1	carrying capacity of Y
β	0.5	flight coefficient
γ	0.45	assimilation coefficient
k	1	social integration coefficient
e	1	exponent in the social advancement term

The next step is calculating the equilibria. This happens just as in the previous sections by using the command `calcep`.

```
ocEPo=calcep(o,rand(4,10),opt); d=isadmissible(o,ocEPo,opt);ocEPo(~d)=[];
d=isnegativestate(o,ocEPo);ocEPo(d==1)=[]; ocEPo=uniqueoc(ocEPo,opt);
ocEPo{:}
```

We receive the following equilibra:

```
ans =

  dynprimitive object:

      Coordinates:
             0.18089
             1.122
             0.00050916
             0.0038786

      Arc identifier:  1
      Linearization:   [4x4 double]


ans =
```

```
  dynprimitive object:

      Coordinates:
             0.9295
             1.0619
             -2.0909e-005
             0.0041755

      Arc identifier:  1
      Linearization:   [4x4 double]


ans =

  dynprimitive object:

      Coordinates:
             0.54914
             1.3985
             -0.12641
             0.010077

      Arc identifier:  1
      Linearization:   [4x4 double]
```

Before continuing our analysis we will check the stability of the first and the second equlibrium by typing `eig(ocEPo{1})` and `eig(ocEP{2})`. We get the following results:

```
ans =

    2.0731
    2.6860
   -2.6360
   -2.0231

ans =

   -2.0416
```

```
  -2.3747
   2.0916
   2.4247
```

In both cases the equilibria exhibit again two-dimensional stable manifolds because the number of eigenvalues ξ satisfying $Re\xi < 0$ is two. We had the same cases in the previous modifications.

Now we will first start a continuation process from the first into the second equilibrium and then do the same vice-versa.

```
initStruct=initoccont('extremal',o,'initpoint',[1 2],ocEPo{1}.
dynVar(1:2,1),ocEPo{2},'IntegrationTime',500);
```

As this continuation process works out we will now solve the BVP problem by typing the command `[solo solno]=occont(o,initStruct,opt);`. The toolbox does not manage to finish the calculation, this is always a sign for possible DNSS points. We stop the calculation and store the results with the command `o=store(o)`. Then we go through the same process for the other direction.

```
initStruct=initoccont('extremal',o,'initpoint',[1 2],ocEPo{2}.
dynVar(1:2,1),ocEPo{1},'IntegrationTime',500);

[solo solno]=occont(o,initStruct,opt);
```

Again the toolbox does not manage to finish the calculation, so we stop it again and store the results.

Now we want to find out if there are DNSS points, so we type the command `pt=finddnss(o,1,2)`. And we get the following result:

```
pt =

    0.2669
    1.1151
```

So we can continue our analysis as there is indeed a DNSS point. In order to make it easier to work with the results in the following we save them in `ocEx` by typing `ocEx=extremalsol(o)`.

Now we start a continuation process from the DNSS point to the first and the second equilibrium and solve the corresponding BVPs. It is important to remember storing all the results.

```
initStruct=initoccont('extremal',o,'initpoint',[1 2],pt,ocEx{1},
'IntegrationTime',500);

[solo solno]=occont(o,initStruct,opt);

o=store(o)

initStruct=initoccont('extremal',o,'initpoint',[1 2],pt,ocEx{2},
'IntegrationTime',500);

[solo solno]=occont(o,initStruct,opt);

o=store(o)
```

We save again all the results and then put the two solution paths together.

```
ocEx=extremalsol(o)

indiff=[ocEx{3} ocEx{4}]
```

Then we need one more path where we fix one of the coordinates, that should be close to the DNSS point. Here we choose the second coordinate and set it equal to 1.12. Then we solve the BVP again and store the results.

```
initStruct1=initoccont('dnss',o,'initpoint',2,1.12,indiff)

[solo solno]=occalc(o,initStruct1,opt);

o=store(o)
```

For the purpose of receiving the diagram (Figure 7.14), we need to save the two paths separately. Then we start the continuation processes from the DNSS point to different points in order to get the indifference curve. Here we chose $Y = 0, 2$.

```
sol1=solo(1).octrajectory

sol2=solo(2).octrajectory

initStruct1=initoccont('dnss',o,'initpoint',2,0,solo)
```

```
[solo1 solno1]=occont(o,initStruct1,opt);

o=store(o)

initStruct1=initoccont('dnss',o,'initpoint',2,2,solo)

[solo1 solno1]=occont(o,initStruct1,opt);

o=store(o)

dnss=o.ocResults.DNSSCurve

plot(sol1.dynVar(1,:),sol1.dynVar(2,:),'r')
hold all
plot(sol2.dynVar(1,:),sol2.dynVar(2,:),'k')
hold all
plot(dnss{3}.dynVar(1,:),dnss{3}.dynVar(2,:),'b')
hold all
plot(dnss{4}.dynVar(1,:),dnss{4}.dynVar(2,:),'b')
```

Time Paths

Another interesting result are diagrams where the control u is displayed as function of time t. For this purpose we type the following commands:

```
plot(sol1.t.*sol1.timeintervals,control(o,sol1),'r')
plot(sol2.t.*sol2.timeintervals,control(o,sol2),'k')
```

So we get two diagrams that show the control u of the two solution paths, leading to the two optimal solutions that we have found. As we see below, they are quite different from each other. The first one (Figure 7.15), belonging to the first equilibrium needs very high u in the beginning. In the diagram it looks like u is varying a lot but in fact it stays between 0.35 and 0.36 at all times. u is first decreasing very fast, then increasing again a bit and finally becoming almost completely constant after approximately $t = 3.5$.

The diagram of the second equilibrium qualitively looks the same (Figure 7.16). u

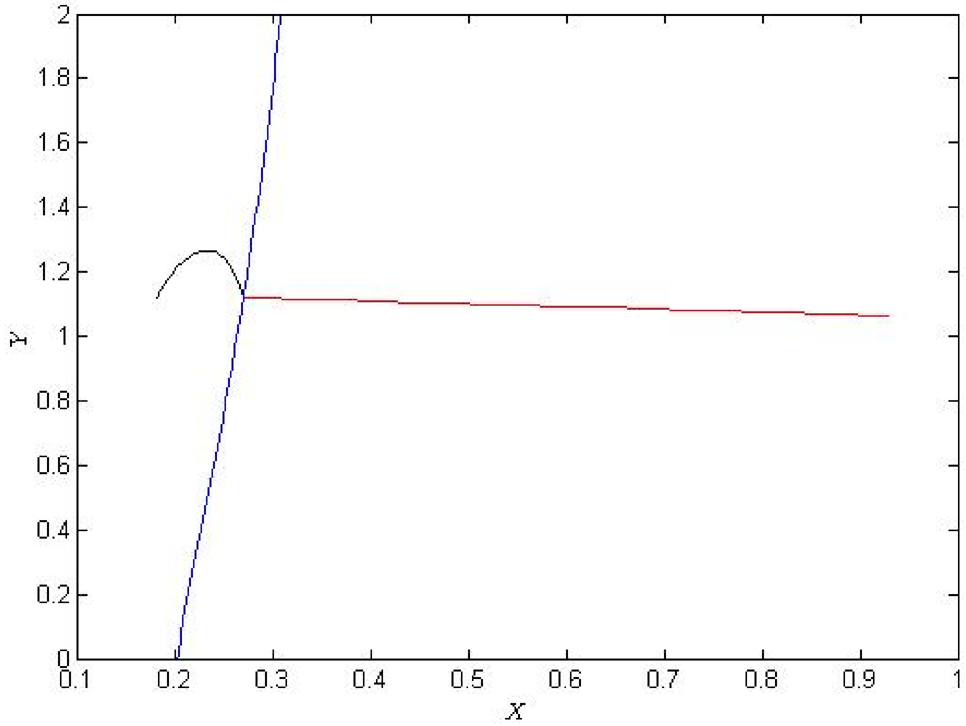

Figure 7.14: DNSS curve for $\rho_X = 0.0005$

is also very high in the beginning, starting at approximately $u = 1.15$, and decreasing fast as in the first diagram. On the other hand it is not increasing again and stays very low. Again u stays almost constant after $t = 3.5$ but it is a lot lower (around 0.2).

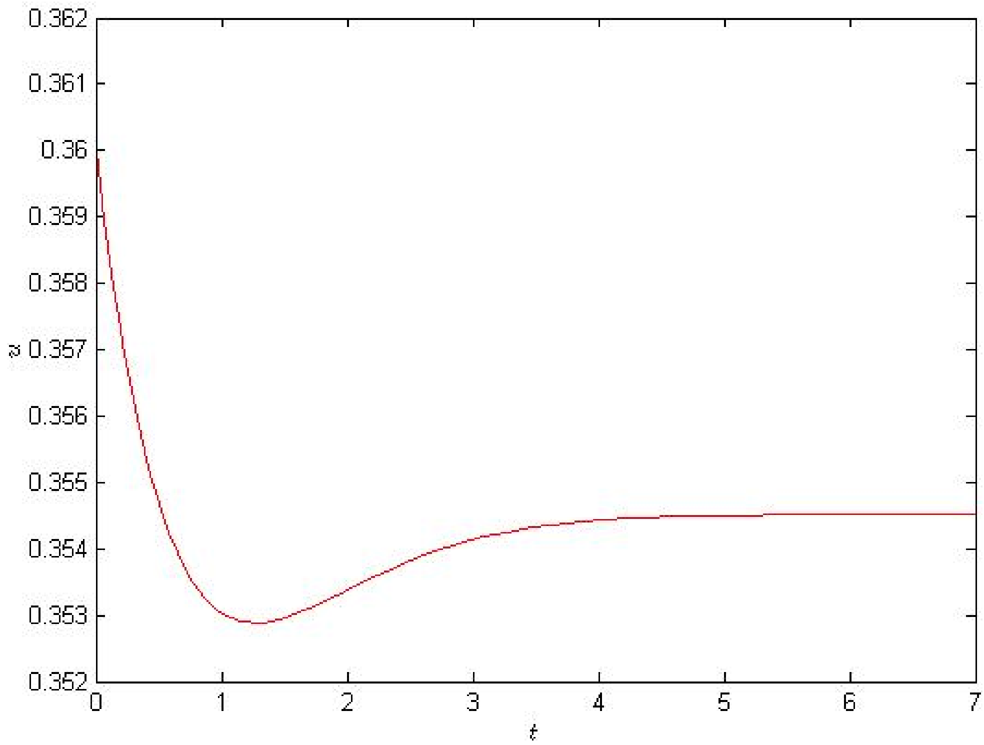

Figure 7.15: Displaying the control u over time towards the first equilibrium

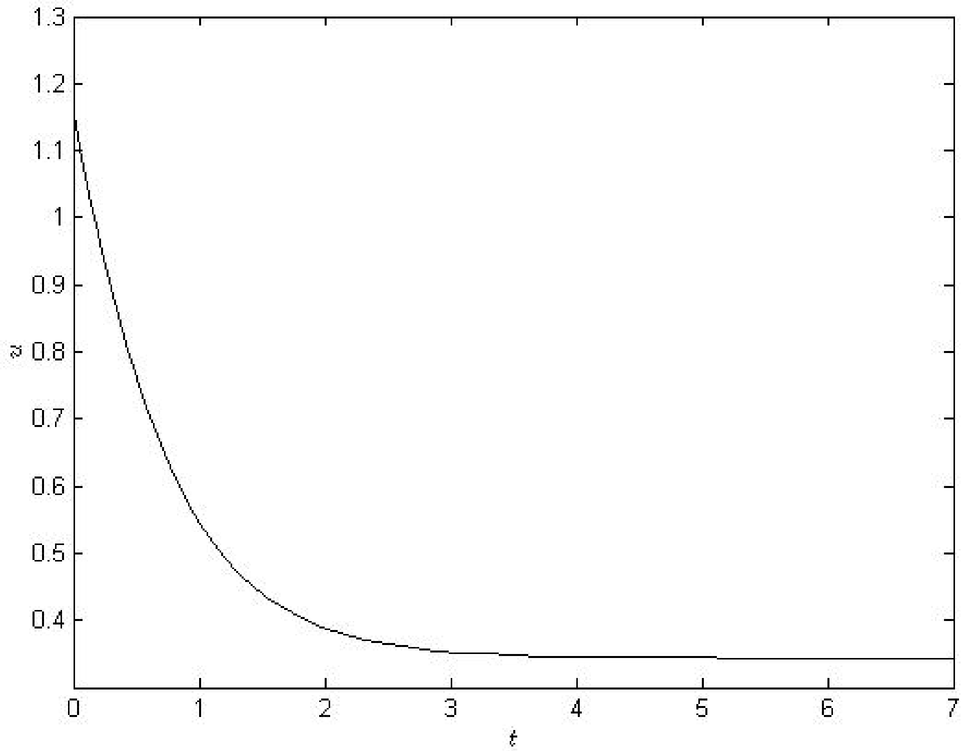

Figure 7.16: Displaying the control u over time towards the second equilibrium

CHAPTER 8

Summary and Conclusions

A central question that the reader may ask now is what kind of impact the modifications of the basic model actually have. Regarding the optimal solution there have not been any significant changes. The original idea was modifying the model that we already had which was analyzed by Reka Horvath (2011) to see how the equilibria develop, and if multiple optimal solutions appear. This has not happened. Of course the equilibria all develop into various directions, but after all the optimal solution is always the one which is the most similar to the optimal solution of the basic model.

Regarding the α Model it has been interesting to see that we can find two different values of α, which both deliver the same value for $\hat{Y}$. When we looked at the time paths we also saw that the two diagrams we created for the second value of α were similar to the corresponding diagrams for the first value of α. It is quite likely that the other two diagrams would also have been similar.

In the cases of the maximum value of α and also in the case $\alpha = 1.5604$, we always had troubles calculating the solution path when the initial state $\hat{X}/2$ was involved. As we said in the previous chapter, X seems to be of big importance for the model. This could also explain why the calculations become more complicated for smaller values of X.

Regarding the β Model, it was interesting to see that there are no big changes at all if β is kept between 0 and 0.5. The most interesting result in that chapter was creating a DNSS curve by modifying the parameters even more. By changing the parameter ρ_X we finally managed to find multiple optimal solutions that allowed us to continue our analysis in order to create a DNSS curve. As mentioned in that chapter we saw that the time paths were qualitatively the same.

List of Figures

List of Tables

APPENDIX A

Appendix: Further Analysis of the α Model

A.1 The Case $\alpha = 0.2$

As there were no big changes in the model for $\alpha = 0.1$, I decided to take a look at the case $\alpha = 0.2$. First we will examine to which extent the equlibria change and then take a closer look at them to see which one is the optimal solution.

So first of all we need to change the parameter. We do this by using the following command: `n=changeparameter(n,'alpha',0.2)`. We can always check the parameters by using the command `showparameter(n)`. So in our case we have the parameters as displayed in Table A.1.

After defining the new model we initialise it and follow the same prodecure as in the original model. So first of all we need to find the equilibria of the model. As in the original model we get the solutions by using the following commands:

```
ocEPn=calcep(n,rand(4,10),opt); c=isadmissible(n,ocEPn,opt);ocEPn(~c)=[];
c=isnegativestate(n,ocEPn);ocEPn(c==1)=[]; ocEPn=uniqueoc(ocEPn,opt);
ocEPn{:}
```

We receive these equilibria:

```
ans =

  dynprimitive object:
```

Table A.1: α Model parameters.

Parameter	Value	Description
r	0.05	discount rate
ρ_X	0.02	objective function coefficient on X
ρ_Y	0.01	objective function coefficient on Y
σ	0.01	weight on objective function control terms
c	2	program cost coefficient
a	2	maximal growth rate at $X = 0$
b	2	maximal growth rate at $Y = 0$
d	1	carrying capacity of Y
α	0.2	rich to poor coefficient
β	0.5	flight coefficient
γ	0.45	assimilation coefficient
k	1	social integration coefficient
e	1	exponent in the social advancement term

```
      Coordinates:
             0.95952
             1.0886
             0.010363
             0.0044036

      Arc identifier:  1
      Linearization:   [4x4 double]


ans =

  dynprimitive object:

      Coordinates:
             0.42477
             1.3771
             -0.063272
             0.0023492

      Arc identifier:  1
      Linearization:   [4x4 double]
```

```
ans =

  dynprimitive object:

      Coordinates:
             0.12448
             1.1935
             -0.011971
             0.0035192

      Arc identifier:  1
      Linearization:   [4x4 double]
```

As we can easily see, the first equilibrium is similar to the optimal solution that we had in the original model, which was $\hat{X} = 1.0485$, $\hat{Y} = 1.0141$, $\hat{\lambda}_1 = 0.009113$ and $\hat{\lambda}_2 = 0.0049762$. $\hat{X}$ and $\hat{\lambda}_2$ have decreased whereas $\hat{Y}$ and $\hat{\lambda}_1$ have increased. Now we suspect again that the first equilibrium is the optimal solution, but in order to be sure we have two different ways of proving it. To analyse the first and the second equilibrium, we can start a continuation process from the state of the first equilibrium into the second equilibrium and vice versa by using:

```
initStruct1=initoccont('extremal',n,'initpoint',[1 2],ocEPn{2}.
dynVar(1:2,1),ocEPn{1},'IntegrationTime',500);
```

If the continuation processes is successful, the corresponding path is superior and the minor stable path can be excluded. In the underlying case, the above initialized continuation is successful, i.e., the second equilibrium can be excluded.
As mentioned above we will also prove in another way that the second equilibrium can be excluded. For this we use the Hamiltonian function. But before doing that we have to solve the BVP. The toolbox solves by

```
[soln solnn]=occont(n,initStruct1,opt);
```

To retrieve a result (already stored in `n`) one can use `ocEPn=equilibrium(n)` for retrieving the elements of the fields of the equilibrium and `ocExn=extremalsol(n)` for retrieving the elements of the field corresponding to the stable path. They are stored in `ocResults` among other calculated elements. Note that one can check all the stored calculations made before by calling `n.ocResults`.

Now that we have calculated the BVP we can go back to the Hamiltonian. The next task is to compare the Hamiltionian of the second equilibrium with the Hamiltonian of the BVP. For the first Hamiltonian we use the command `hamiltonian(n,ocEPn{2})` and get the following result

```
ans =

    0.0091    0.0091
```

For the second Hamiltonian we use the command `hamiltonian(n,soln)`and get the following results

```
ans =

  Columns 1 through 10

    0.0310    0.0310    0.0310    0.0310    0.0311    0.0311    0.0311
0.0312    0.0312    0.0312

  Columns 11 through 20

    0.0312    0.0313    0.0313    0.0313    0.0313    0.0313    0.0313
0.0313    0.0313    0.0313

  Columns 21 through 30

    0.0313    0.0313    0.0313    0.0313    0.0313    0.0313    0.0313
0.0313    0.0313    0.0313

  Columns 31 through 40

    0.0313    0.0313    0.0313    0.0313    0.0313    0.0313    0.0313
0.0313    0.0313    0.0313

  Column 41

    0.0313
```

As we see the first Hamiltonian is clearly smaller than the second one. The rule is that if the command `hamiltonian(n,ocEPn{2})`delivers a smaller result than the command `hamiltonian(n,soln)` we know that the second equilibrium is not an optimal solution.

Thus we stick with the suspection that the first equilibrium is the optimal solution.

Now we go through the same procedure using the third equlibrium. So we re-use the commands from above and eventually get the same results, namely that the first equilibrium is indeed the optimal solution (Figure A.1).

The next step is having a look at the phase diagram of the optimal solution (Figure A.1). We will have X on the x-axis and Y on the y-axis and use the following starting points:

$(\hat{X}/2, \hat{Y}/2) = (0.4798, 0.5443)$
$(2\hat{X}, 2\hat{Y}) = (1.9190, 2.1772)$
$(\hat{X}/2, 2\hat{Y}) = (0.4798, 2.1772)$
$(2\hat{X}, \hat{Y}/2) = (1.9190, 0.5443)$

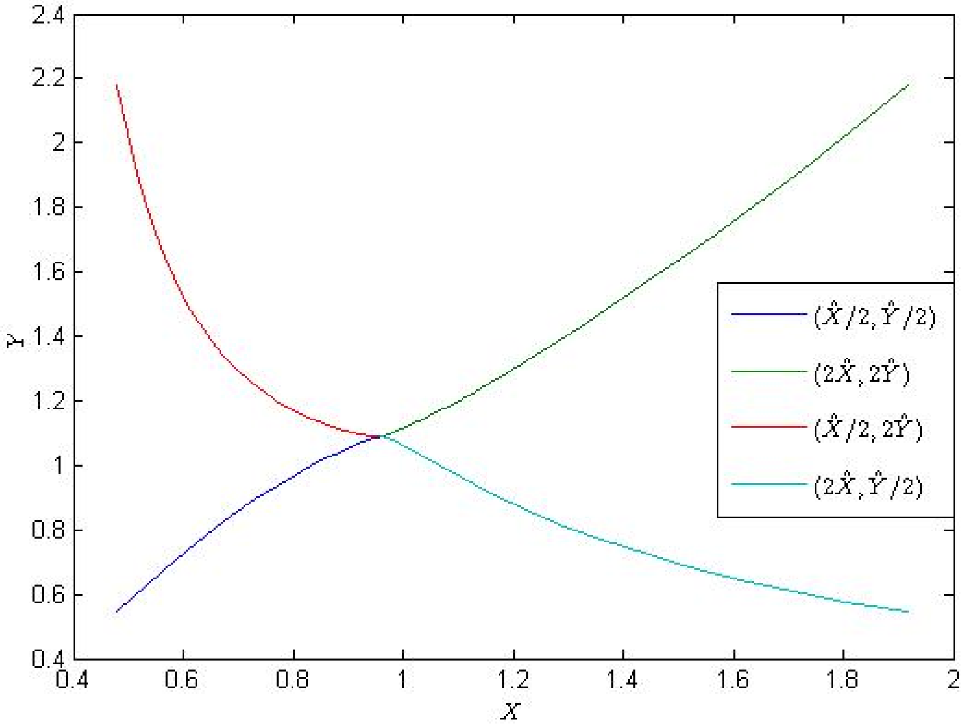

Figure A.1: X in relation to Y

A.2 The Case $\alpha = 0.5$

After having analyzed the case $\alpha = 0.2$, we will now examine the case where α equals 0.5 and see how and if the equilibria change. Furthermore we will find out which of the delivered equilibria is the optimal solution.

Again the first step is to change the parameter, which we do by using the following command: `n=changeparameter(n,'alpha',0.5)`. We can always check the parameters by using the command `showparameter(n)`. So in our case we have the parameters from Table A.2. After defining the new model we initialise it and follow the same

Table A.2: α Model parameters.

Parameter	Value	Description
r	0.05	discount rate
ρ_X	0.02	objective function coefficient on X
ρ_Y	0.01	objective function coefficient on Y
σ	0.01	weight on objective function control terms
c	2	program cost coefficient
a	2	maximal growth rate at $X = 0$
b	2	maximal growth rate at $Y = 0$
d	1	carrying capacity of Y
α	0.5	rich to poor coefficient
β	0.5	flight coefficient
γ	0.45	assimilation coefficient
k	1	social integration coefficient
e	1	exponent in the social advancement term

prodecure as before. So first of all we need to find the equilibria of the model. As in the original model, we get the solutions by using the following commands:

```
ocEPn=calcep(n,rand(4,10),opt); c=isadmissible(n,ocEPn,opt);ocEPn(~c)=[];
c=isnegativestate(n,ocEPn);ocEPn(c==1)=[]; ocEPn=uniqueoc(ocEPn,opt);
ocEPn{:}
```

We receive these equilibria:

```
ans =

  dynprimitive object:

      Coordinates:
             0.82399
             1.1653
```

```
            0.012628
            0.0039357

      Arc identifier:  1
      Linearization:   [4x4 double]


ans =

  dynprimitive object:

      Coordinates:
            0.19633
            1.2447
            -0.020523
            0.0032359

      Arc identifier:  1
      Linearization:   [4x4 double]


ans =

  dynprimitive object:

      Coordinates:
            0.28352
            1.2986
            -0.033235
            0.0029212

      Arc identifier:  1
      Linearization:   [4x4 double]
```

Again we suspect the first equilibrium to be the optimal solution of the model because of its similarity to the previous optimal solution and the optimal solution of the original model. As a reminder, the optimal solution of the $\alpha = 0.2$ model was $\hat{X} = 0.95952$, $\hat{Y} = 1.0886$, $\hat{\lambda}_1 = 0.010363$ and $\hat{\lambda}_2 = 0.0044036$. So again we find that $\hat{X}$ and $\hat{\lambda}_2$ have decreased whereas $\hat{Y}$ and $\hat{\lambda}_1$ have increased. Of course we have to prove again that the first equilibrium is indeed the optimal solution. We follow the same procedure as above and use two different ways of proving it. To analyze the first and the second equilibrium, we can start a continuation process from the state of the first equilibrium into the second equilibrium and vice versa by using:

```
initStruct1=initoccont('extremal',n,'initpoint',[1 2],ocEPn{2}.
dynVar(1:2,1),ocEPn{1},'IntegrationTime',500);
```

If the continuation processes is successful, the corresponding path is superior and the minor stable path can be excluded. In the underlying case, the above initialized continuation is successful, i.e., the second equilibrium can be excluded.

As mentioned above we will also prove in another way that the second equlibrium can be excluded. For this we use the Hamiltonian function. But before doing that we have to solve the BVP. The toolbox solves by

```
[soln solnn]=occont(n,initStruct1,opt);
```

To retrieve a result (already stored in `n`) one can use `ocEPn=equilibrium(n)` for retrieving the elements of the fields of the equilibrium and `ocExn=extremalsol(n)` for retrieving the elements of the field corresponding to the stable path. They are stored in `ocResults` among other calculated elements. Note that one can check all the stored calculations made before by calling `n.ocResults`.

Now that we have calculated the BVP we can go back to the Hamiltonian. The next task is to compare the Hamiltionian of the second equilibrium with the Hamiltonian of the BVP. For the first Hamiltonian we use the command `hamiltonian(n,ocEPn{2})` and get the following result

```
ans =

    0.0153    0.0153
```

For the second Hamiltonian we use the command `hamiltonian(n,soln)` and get the following results

```
ans =

  Columns 1 through 10
```

```
    0.0289    0.0289    0.0290    0.0290    0.0291    0.0291    0.0291
0.0292    0.0292    0.0292

  Columns 11 through 20

    0.0293    0.0293    0.0293    0.0293    0.0293    0.0293    0.0293
0.0293    0.0293    0.0293

  Columns 21 through 30

    0.0293    0.0293    0.0293    0.0293    0.0293    0.0293    0.0293
0.0293    0.0293    0.0293

  Columns 31 through 36

    0.0293    0.0293    0.0293    0.0293    0.0293    0.0293
```

As we see, the first Hamiltonian is clearly smaller than the second one. The rule is that if the command `hamiltonian(n,ocEPn{2})` delivers a smaller result than the command `hamiltonian(n,soln)` we know that the second equilibrium is not an optimal solution. Thus we know for sure that the first equilibrium compared to the second equilibrium is the optimal solution.

Now we go through the same procedure using the third equlibrium. So we re-use the commands from above and eventually get the same results, namely that the first equilibrium is indeed the optimal solution.

The next step is having a look at the phase diagram of the optimal solution (Figure A.2). We will have X on the x-axis and Y on the y-axis and use the following starting points::

$(\hat{X}/2, \hat{Y}/2) = (0.4120, 0.5827)$
$(2\hat{X}, 2\hat{Y}) = (1.6480, 2.3306)$
$(\hat{X}/2, 2\hat{Y}) = (0.4120, 2.3306)$
$(2\hat{X}, \hat{Y}/2) = (1.6480, 0.5827)$

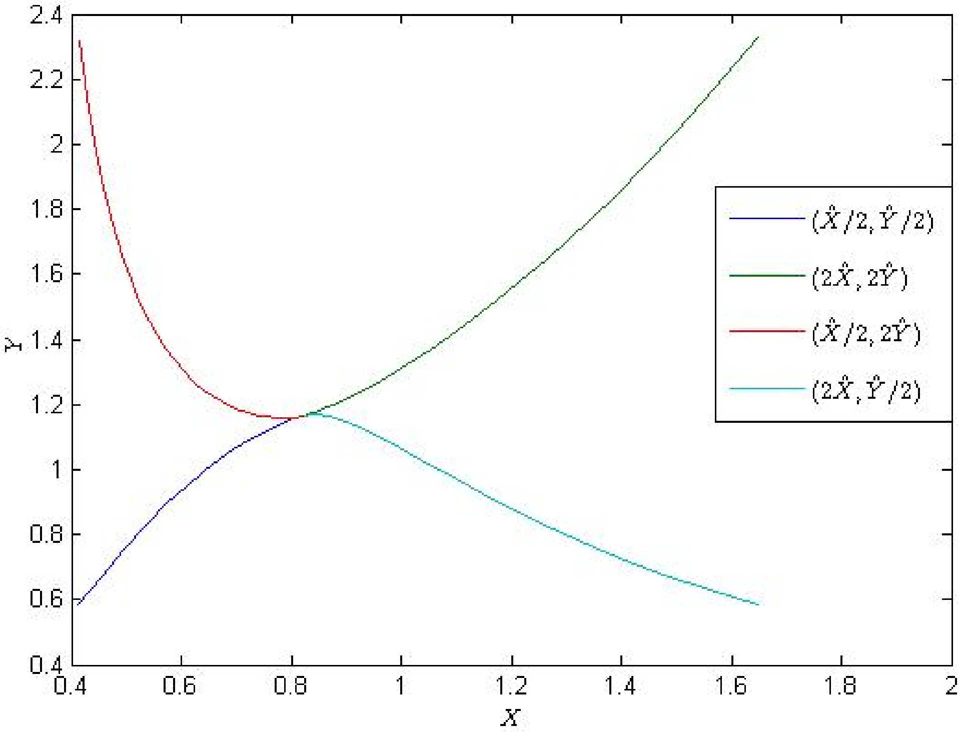

Figure A.2: X in relation to Y

A.3 The Case $\alpha = 1$

There is an interesting development if we continue to increase α, namely that Matlab delivers only one equilibrium starting from the value 0.6 for α. Luckily, that equilibium is exactly the one that fits with the optimal solution of the previous cases. This time I took a bigger step and had a look at the case $\alpha = 1$. Therefore the only equilbrium and thus the optimal solution looks as follows:

```
ans =

  dynprimitive object:

      Coordinates:
             0.60318
             1.2197
             0.018174
             0.0036587

      Arc identifier:  1
      Linearization:   [4x4 double]
```

So again we find that $\hat{X}$ and $\hat{\lambda}_2$ have decreased whereas $\hat{Y}$ and $\hat{\lambda}_1$ have increased. As this is our only equilibrium, there is no need to continue with further analysis. So we take a look at the phase diagram, which shows us how X and Y develop in relation to each other (Figure A.3). We use the following starting points::

$(\hat{X}/2, \hat{Y}/2) = (0.3016, 0.6099)$
$(2\hat{X}, 2\hat{Y}) = (1.2064, 2.4394)$
$(\hat{X}/2, 2\hat{Y}) = (0.3016, 2.4394)$
$(2\hat{X}, \hat{Y}/2) = (1.2064, 0.6099)$

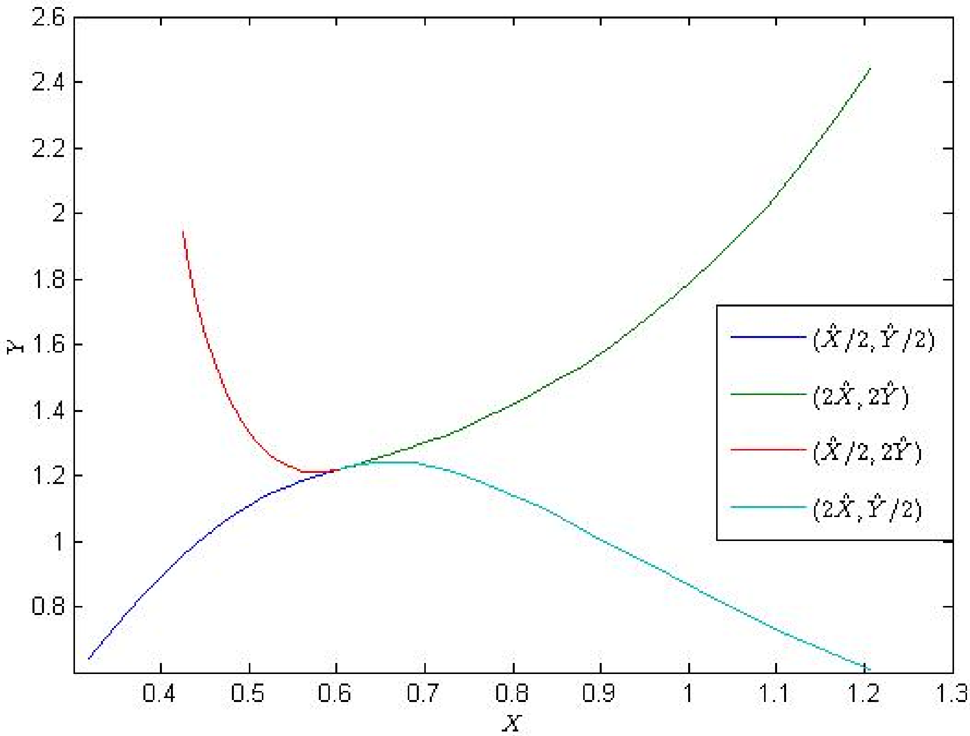

Figure A.3: X in relation to Y

Bibliography

R. Bernstein and T. Edwards. An older and more diverse nation by midcentury. *U.S.Census Bureau*, August, 14 2008. URL http://www.census.gov/newsroom/releases/archives/population/cb08-123.html.

J.R. Betts and R.W. Fairlie. Does Immigration Induce 'Native Flight' from Public Schoolsinto Private Schools? *Elsevier, Journal of Public Economics Journal*, 83:987-1012, 2003.

Bureau of Labor Statistics. Employment situation summary. In *Economic News Release*, August, 2011. URL http://www.bls.gov/news.release/empsit.nr0.htm. September 18, 2011.

Bureau of Labor Statistics. Labor force statistics from the current population survey. In *Unemployment Rate*, August, 2011. URL http://www.bls.gov/cps/. September 18, 2011, Vienna.

J.P. Caulkins, G. Feichtinger, D. Grass, M. Johnson, G. Tragler, and Y. Yegorov. Placing the poor while keeping the rich in their place: Separating strategies for optimally managing residential mobility and assimilation. *Demographic Research*, 13(1):1-34, 2005a.

J.P. Caulkins, G. Feichtinger, D. Grass, and G. Tragler. A model of moderation: Finding Skiba points on a slippery slope. *Central European Journal of Operations Research*, 45-64(1):13, 2005b.

J.P. Caulkins, G. Feichtinger, G. Tragler, D. Behrens, and S. Bushway. Population dynamics model. Memo #26, 1999.

C. T. Clotfelter. Are whites still feeing? racial patterns and enrollments shifts in urban public school. *Journal of Policy Analysis and Management*, 20(2):199-221, 2001.

C. T. Clotfelter. After Brown: The Rise and Retreat of School Desegregation. *Princeton University Press.*, 2004.

D.W. Dechert and K. Nishimura. A complete characterization of optimal growth paths in an aggregated model with nonconcave production function. *Journal of Economic Theory*, 31:332-354, 1983.

C. DeNavas-Walt, B.D. Proctor, and J.C. Smith. Income, poverty, and health insurance coverage in the united states: 2008. *U.S. Census Bureau*, 2008.

H. Elhassan, J. Feins, J. Goering, H.J. Holin, J. Kraft, and D. McInnis. Moving to opportunity for fair housing demonstration program: Current status and initial findings. Technical report, *U.S. Department of Housing and Urban Development Office of Policy Development and Research*, 1999. URL http://www.huduser.org/publications/pdf/mto.pdf.

I.G. Ellen. Sharing america's neighborhoods: The prospects for stable, racial integration. Technical report, *Harvard University Press*, 2000.

R.W. Fairlie. Private schools and 'latino flight' from black schoolchildren. *Demography*, 39(4):655674, 2002.

G. Feichtinger and R.F. Hartl. Optimale Kontrolle ökonomischer Prozesse: Anwendungen des Maximumprinzips in den Wirtschaftswissenschaften. *Walter de Gruyter, Berlin*, 1986. URL http://www.univie.ac.at/bwl/prod/hp/hartl/FeiHaBook/OCOP.pdf.

M. Fujita. Urban Economic Theory: Land Use and City Size. *Cambridge University Press*, 1989.

F. Gaschet and N. Gaussier. Urban segregation and labour markets within the bordeaux metropolitan area: an investigation of the spatial friction. Technical report, *Université Montesquieu-Bordeaux IV*, 2004.

D. Grass. OCMat: A MATLAB package for the analysis of optimal control problems. *Operations Research and Control Systems (ORCOS), Vienna University of Technology*, 2010-2011. URL http://orcos.tuwien.ac.at/research/ocmat_software.

D. Grass, J.P. Caulkins, G. Feichtinger, G. Tragler, and D.A. Behrens. Optimal Control of Nonlinear Processes: With Applications in Drugs, Corruption, and Terror. *Springer Verlag, Berlin*, 2008.

D. Grass and A. Seidl. OCMat Manual, 2008.

D. Grass and G. Tragler. Optimal dynamic management of the population mix. *Central European Journal of Operations Research*, 18(4):539551, 2010.

V. Henderson. The sizes and types of cities. *American Economic Review*, 64:640-656, 1974.

R. Horvath, Optimal Dynamic Management of the Population Mix: Application of the OC Matlab Toolbox. *Vienna University for Technology*, 5-30, 2011

F. Kemper. Restructuring of housing and ethnic segregation: Recent developments in Berlin. *Urban Studies*, 35(10):1765-1789, October 1998.

S.E. Mayer and C. Jencks. Poverty and the distribution of material hardship. *Journal of Human Resources*, 24(1):88-114, 1989.

G. Meen, K. Gibb, J. Goody, T. McGrath, and J. Mackinnon. Economic segregation in England: causes, consequences and policy, volume ISSN 0958-3084. *Joseph Rowntree Foundation*, December 2005.

Die Presse. Österreich ist EU-Land mit niedrigster Arbeitslosigkeit. August 31, 2011. URL `DiePresse.com`. > Wirtschaft > National.

T. Schelling. Dynamic models of segregation. Journal of Mathematical Sociology, Vol. 1, 1971. Abbreviated version appeared as “Models of Segregation”, in *The American Economic Review*, Vol. LIX, No. 2, May 1969.

B. Schindler Rangvid. Living and learning separately? ethnic segregation of school children in copenhagen. *Urban Studies*, 44(7):1329-1354, June 2007.

S.P. Sethi. Nearest feasible paths in optimal control problems: Theory, examples, and counterexamples. *Journal of Optimization Theory and Applications*, 23:563-579, 1977. 4, Dec. 1977.

S.P. Sethi. Optimal advertising policy with the contagion model. *Journal of Optimization Theory and Applications*, 29:615-627, 1979. 4, Dec. 1979.

A.K. Skiba. Optimal growth with a convex-concave production function. *Econometrica*, 46:527-539, 1978.

P. Starr. A new deal of their own. *American Prospect*, February 25, 2008.

UNICEF. Child poverty in perspective: An overview of child well-being in rich countries. *Innocenti Research*, Report Card 7, 2007. URL `http://www.unicef-irc.org/publications/pdf/rc7_eng.pdf`.

United States Census Bureau. Annual estimates of the population of metropolitan and micropolitan statistical areas: April 1, 2000 to july 1, 2008 (cbsa-est2008-01). In *Population Division*, March 19, 2009. URL `http://www.census.gov/popest/metro/CBSA-est2008-annual.html`. September 18, 2011, Vienna.

United States Census Bureau. Resident population by sex, race, and hispanic origin status: 2000 to 2009. In *Statistical Abstract of the United States*, 2011a. URL `http://www.census.gov/compendia/statab/2011/tables/11s0006.pdf`.

United States Census Bureau. U.S. Popclock Projection. In *U.S. & World Population Clocks*, 2011b. URL `http://www.census.gov/population/www/popclockus.html`. September 17, 2011 20:00, Vienna.

G. Verdugo. Public housing and residential segregation of immigrants in france, 1968-1999. *IZA Discussion Paper*, No. 5456, 2011.