CHAPTER ELEVEN

Computing All Solutions to Polynomial Equations in Economics

Felix Kubler*, Philipp Renner†, and Karl Schmedders‡

*Department of Banking and Finance, University of Zurich, and Swiss Finance Institute, Switzerland
†Hoover Institution, Stanford, CA, USA
‡Department of Business Administration, University of Zurich, and Swiss Finance Institute, Switzerland

Contents

1. Introduction 600
2. Gröbner Bases and Polynomial Equations 602
   2.1 What Is a Gröbner Basis? A Brief Introduction 603
      2.1.1 A Formal Definition of Gröbner Bases 605
      2.1.2 Elimination Ideals and the Shape Lemma 608
      2.1.3 Buchberger’s Algorithm 609
      2.1.4 Computing Gröbner Bases with Computer Algebra Systems 611
   2.2 Using Gröbner Bases to Solve Polynomial Systems 614
      2.2.1 Root Count for Univariate Polynomials 614
      2.2.2 Sufficient Conditions for the Shape Lemma 615
      2.2.3 What If the Shape Lemma Does Not Apply? 616
      2.2.4 Finding All Solutions with Computer Algebra Systems 617
      2.2.5 Parameterized Gröbner Bases 618
      2.2.6 Parameterized Shape Lemma with Computer Algebra Systems 620
3. Applying Gröbner Bases to Economic Models 622
   3.1 A Bertrand Game 622
      3.1.1 Polynomial Equilibrium Equations 624
      3.1.2 Solving the System with SINGULAR 624
      3.1.3 The Manifold of Positive Solutions 625
   3.2 A Simple Walrasian Exchange Economy 625
      3.2.1 Polynomial System and Equilibria 626
      3.2.2 Finding All Equilibria with SINGULAR 627
4. All-Solution Homotopy Methods 631
   4.1 Brief Introduction to All-Solutions Homotopies 631
      4.1.1 Mathematical Background 632
      4.1.2 All Roots of Univariate Polynomials 633
      4.1.3 Multivariate Systems of Polynomial Equations 637
   4.2 Advanced Features of All-Solution Homotopy Methods 639
      4.2.1 Homogenization and Projective Space 639
      4.2.2 The m-Homogeneous Bezout Number 643
      4.2.3 Parameter Continuation Homotopy 644
5. Applying Homotopy Methods 645
   5.1 Software 645
1. INTRODUCTION

Many economic models, both those with strategic interactions and those with competitive markets, have multiple equilibria. Recent examples of such models include, among many others, Bodenstein (2010) in international macroeconomics, Foerster et al. (2013) in Markov-switching DSGE models, Besanko et al. (2010), Doraszelski and Satterthwaite (2010), Borkovsky et al. (2012), and Iskhakov et al. (2013) in industrial organization, Baliga and Sjöström (2004) in an arms race game under incomplete information with cheap talk, and Kubler and Schmedders (2010a) in general equilibrium models. Of course, equilibrium multiplicity has long been a well-known feature of many normal-form games; see Sturmfels (2002) and Datta (2010). The presence of multiple equilibria complicates the analysis of model predictions, policy experiments, and structural estimation. In fact, equilibrium multiplicity poses a serious threat to the validity of such analyses. This threat is particularly acute if not all equilibria of the examined model are known. In many applications in economics and finance, we suspect that multiple equilibria may exist, but standard numerical solution methods only search for a single solution.

We argue that in many economic and financial applications all equilibria can be described as solutions to systems of polynomial equations. Recent advances in the field of computational algebraic geometry have resulted in greatly improved methods for finding all solutions to polynomial systems of equations which have also been implemented in software packages. For the purpose of this article, we understand “numerically solving polynomial systems” as computing approximations to all isolated solutions and disregard continua of solutions. We describe two very different solution methods, provide hands-on introductions to available software packages, and determine all equilibria for some economic applications. We focus on what we believe are currently the two most promising solution approaches for economic applications, Gröbner bases methods and all-solution homotopy methods.

In the first half of this article, we discuss how Gröbner bases can be used to solve systems of polynomial equations. This solution method is mostly algebraic in nature...
and does not rely much on numerical analysis. The basic idea is that for a system of polynomials with finitely many zeros, so-called Gröbner bases form an equivalent system that has a triangular form. Based on such a Gröbner basis, we can then determine all values of a single (chosen) variable by numerically solving a univariate polynomial. The values of all other variables in all solutions are then simply polynomial functions of the values of that single chosen variable. The derivation of the Gröbner basis can be performed with standard computer algebra packages, for example, in Mathematica®.1 Alternatively, there are good specialized software packages which are available free of charge, for example, the computer algebra system Singular.2 If all coefficients in the original system of polynomial equations are rational numbers or parameters, then the computations of Mathematica or Singular are exact. Therefore, Gröbner bases can not only be used for a numerical approximation of equilibria, but in fact may allow the proof of theoretical results for the underlying economic model. Under a mild assumption, only the step of solving the univariate equation in the chosen variable requires numerical approximations, for which very efficient and reliable methods exist.

For a basic understanding of the fundamental properties of Gröbner bases, we first need to briefly introduce polynomial ideals and complex varieties. After a formal definition of Gröbner bases, we briefly describe Buchberger’s algorithm for the computation of such bases. Finally we explain how to use a computed Gröbner basis for the numerical calculation of all solutions. We attempt to equip the reader with as little technical background as necessary to understand the main ideas of Gröbner bases methods. Readers who are interested in more mathematical details should consult Cox et al. (2007) or Sturmfels (2002). The main focus of the present article lies on applications of Gröbner bases to solve systems of polynomial equations. We furnish a series of simple examples to illustrate how to apply Mathematica or Singular to solve them. We also provide several economic examples which illustrate how economic problems can often be naturally written as systems of polynomial equations and which illustrate the great usefulness of Gröbner bases for tackling these economic applications.

In the second half of this article, we describe all-solution homotopy methods which provide an alternative approach to computing all solutions to polynomial systems. There are well-known upper bounds on the maximal number of complex solutions of a square polynomial system. The basic idea of an all-solution homotopy method is to start with a generic system \( g(x) = 0 \) whose number of zeros is at least as large as the maximal number of solutions to the equilibrium system \( f(x) = 0 \) and whose zeros are all known. Starting from each solution to \( g(x) = 0 \), the method traces out the path (in complex space) of the solutions to the homotopy equation \( H(x, t) = tg(x) + (1 - t)f(x) \) for increasing \( t \) starting from \( t = 0 \). All solutions to \( H(x, 1) = f(x) = 0 \) can be found in this manner.

1 Mathematica is a registered trademark of Wolfram Research, Inc. In the remainder of this paper we suppress all trademark signs. All our calculations in Mathematica reported in this paper have been performed with Mathematica 8.

2 Singular is available at www.singular.uni-kl.de; see Decker et al. (2012). All our calculations in Singular reported in this paper have been performed with version 3.1.6.
We again provide as little technical background as necessary to understand the main ideas of the all-solution homotopy approach and refer interested readers to Sommese and Wampler (2005) and Sturmfels (2002) for many more details. After providing some basic intuition and outlining the mathematical foundation for the all-solution homotopies, we describe how to use the software package Bertini\(^3\) which features a variety of root-counting methods among its tools. Essentially we can use the solver as a black box and just provide it with the system of polynomial equations in an input file.

Both families of methods, Gröbner bases methods and all-solution homotopies, have their advantages and disadvantages. Homotopy methods are purely numerical methods and are typically able to solve much larger systems of equations, with more variables and polynomials of higher degrees. In addition, the different homotopy paths (starting at the different zeros of the polynomial \(g(x)\)) can be traced separately and thus homotopy methods are naturally parallelizable. While Gröbner bases methods are slower and with the current technology cannot solve systems as large as the homotopy methods, they also offer some advantages. Since we can compute Gröbner bases for parameterized systems, we can use them to prove theoretical results on the number of equilibria for an economic or financial model. In addition, since we can compute Gröbner bases exactly in many applications, we have fewer numerical difficulties in the computation of all solutions.

The part on Gröbner bases methods in this paper is a greatly expanded version of Kubler and Schmedders (2010a,b) and relies heavily on the textbooks Cox et al. (2007, 1998), and Sturmfels (2002). The part on homotopy methods draws heavily from Judd et al. (2012) and the textbook Sommese and Wampler (2005).

The remainder of the article is organized as follows. Section 2 provides an introduction to Gröbner bases. A reader who is mostly interested in economic applications may want to focus on Section 2.2. In Section 3, we describe two economic examples and illustrate in detail how to solve them with Gröbner bases methods. Section 4 provides a brief introduction to all-solution homotopies and explains how homotopy continuation methods can be used to solve polynomial systems of equations. In Section 5 we revisit some of the economic examples from Section 3 and show how to solve them using homotopy methods. Section 6 concludes.

\section{Gröbner Bases and Polynomial Equations}

For the description of a polynomial \(f\) in the \(n\) variables \(x_1, x_2, \ldots, x_n\) we first need to define monomials. A monomial in \(x_1, x_2, \ldots, x_n\) is a product \(x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_n^{\alpha_n}\) where all exponents \(\alpha_i, \ i = 1, 2, \ldots, n\), are nonnegative integers. It will be convenient to write a monomial as \(x^\alpha \equiv x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_n^{\alpha_n}\) with \(\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{Z}_{+}^n\), the set of nonnegative integer vectors of dimension \(n\). A polynomial is a linear combination of finitely many

\(^3\) Bertini is available at \url{http://www3.nd.edu/~sommese/bertini/}. All our calculations in Bertini reported in this paper have been performed with version 1.3.1.
monomials with coefficients in a field $\mathbb{K}$. We can write a polynomial $f$ as

$$f(x) = \sum_{\alpha \in S} a_\alpha x^\alpha, \quad a_\alpha \in \mathbb{K}, \quad S \subset \mathbb{Z}_+^n \text{ finite.}$$

The integer $\deg(f) = \max\{\|\alpha\|_1 \mid a_\alpha \neq 0\}$ is called the degree of $f$, if $f \neq 0$. If $f = 0$ then we set $\deg(f) = -1$.

We denote the collection of all polynomials in the variables $x_1, x_2, \ldots, x_n$ with coefficients in the field $\mathbb{K}$ by $\mathbb{K}[x_1, \ldots, x_n]$, or, when the dimension is clear from the context, by $\mathbb{K}[x]$. The set $\mathbb{K}[x]$ is called a polynomial ring. In this paper we do not need to consider arbitrary fields of coefficients but instead we can focus on three commonly used fields. These are the field of rational numbers $\mathbb{Q}$, the field of real numbers $\mathbb{R}$, and the field of complex numbers $\mathbb{C}$. Polynomials over the field of rational numbers are computationally convenient since modern computer algebra systems (such as SINGULAR or MATHEMATICA) perform exact computations over the field $\mathbb{Q}$. Economic parameters are typically real numbers and thus force us to consider $\mathbb{R}[x]$. In some parts of the mathematical theory we need to consider the polynomial ring $\mathbb{C}[x]$.

Since we want to solve systems of polynomial equations arising in economics and finance, we are primarily interested in the set of real solutions to a square system of polynomial equations, i.e., given $f_1, \ldots, f_n \in \mathbb{K}[x_1, \ldots, x_n]$ we want to find all elements in the (hopefully finite) set $\{x \in \mathbb{R}^n : f_1(x) = \ldots = f_n(x) = 0\}$. For reasons that become clear below, we often need to consider the set of all complex solutions, $\{x \in \mathbb{C}^n : f_1(x) = \ldots = f_n(x) = 0\}$, although, from an economic point of view, we only need the real solutions.

### 2.1 What Is a Gröbner Basis? A Brief Introduction

We define addition and multiplication of polynomials as one might expect: Given a ring of polynomials $\mathbb{K}[x_1, \ldots, x_n] = \mathbb{K}[x]$, for $\mathbb{K} \in \{\mathbb{C}, \mathbb{R}, \mathbb{Q}\}$, we have

$$\sum_{\alpha} a_\alpha x^\alpha + \sum_{\alpha} b_\alpha x^\alpha := \sum_{\alpha} (a_\alpha + b_\alpha)x^\alpha,$$

$$\left(\sum_{\alpha} a_\alpha x^\alpha\right) \cdot \left(\sum_{\beta} b_\beta x^\beta\right) := \sum_{\gamma} \left(\sum_{\alpha + \beta = \gamma} a_\alpha b_\beta\right)x^\gamma.$$

Let $f_1, \ldots, f_k \in \mathbb{K}[x]$, then

$$I = \left\{ f \mid f = \sum_{i=1}^k p_i f_i \text{ for some } p_i \in \mathbb{K}[x] \right\}$$

---

4 In this introduction we mention some fundamental concepts from algebra such as fields, rings, and polynomial rings. The precise definitions of these terms are not necessary to obtain an understanding of solving polynomial systems of equations and are, therefore, omitted. The interested reader may want to refer to Greuel and Pfister (2002) or any introductory textbook to abstract algebra.
is called an **ideal** in \( \mathbb{K}[x] \) generated by the polynomials \( f_1, \ldots, f_k \) and we write \( I = \langle f_1, \ldots, f_k \rangle \). In words, an ideal is the set of all “linear” combinations of the polynomials \( f_1, \ldots, f_k \), where the coefficients in each “linear” combination are themselves polynomials in the polynomial ring \( \mathbb{K}[x] \). The following theorem links finite systems of equations to ideals.

**Theorem 2.1** (Hilbert’s Basis Theorem). Let \( I \subseteq K[x_1, \ldots, x_n] \) be an ideal. Then there exists \( f_1, \ldots, f_m \), such that \( I = \langle f_1, \ldots, f_m \rangle \).

For an ideal \( I \) we denote by \( V(I) \) the (affine) complex variety of \( I \), which is the set of points in complex space where all the elements of \( I \) vanish. That is, independently of the field \( \mathbb{K} \subseteq \mathbb{C} \), we define the complex variety of an ideal, \( I \subseteq \mathbb{K}[x] \), as

\[
V(I) = \{ x \in \mathbb{C}^n : f(x) = 0 \text{ for all } f \in I \}.
\]

A crucial insight is that if \( I \) is generated by \( f_1, \ldots, f_k \), so \( I = \langle f_1, \ldots, f_k \rangle \), then one can easily verify that \( V(I) = \{ x \in \mathbb{C}^n : f_1(x) = \cdots = f_k(x) = 0 \} \). It is easy to see from the definition of the variety that \( V(I) \subseteq \{ x \in \mathbb{C}^n : f_1(x) = \cdots = f_k(x) = 0 \} \). Moreover, the reversed subset relationship follows from the fact that if \( f_1(x) = \cdots = f_k(x) = 0 \) then \( x \) must also be a zero of all combinations of the \( f_1, \ldots, f_k \).

Note that we define complex varieties for the general case of \( k \) polynomials in \( n \) variables. In this paper, we are mostly concerned with the case \( k = n \), that is, with square systems. In particular, we focus on the case where the variety \( V(I) \) has only finitely many elements. In this case, we call both the complex variety \( V(I) \) and the ideal \( I \) zero-dimensional.

If two sets of polynomials generate the same ideal, the zero sets must be identical. To see why this insight is useful consider a simple example. Consider \( f_1(x_1, x_2) = 2x_1^2 + 3x_2^2 - 11, f_2(x_1, x_2) = x_1^2 - x_2^2 - 3 \). What can we say about the set of common zeros of \( f_1 \) and \( f_2 \), \( \{(x_1, x_2) : f_1(x_1, x_2) = f_2(x_1, x_2) = 0\} \)? We can compute the set directly, but it is also easy to see that \( (f_1, f_2) = (x_1^2 - 4, x_2^2 - 1) \). Therefore, we just need to find the zeros to the univariate polynomials \( x_1^2 - 4 = 0 \) and \( x_2^2 - 1 = 0 \).

Obviously, the example is simple and someone who has never heard the term “ideal” can solve it. So the question arises whether this example is perhaps a consequence of a general insight. Is it always possible to find an alternative basis for a polynomial system that can be solved more easily? It turns out that the **Gröbner basis under lexicographic monomial order** is such a good basis.

Before turning to the formal definition of a Gröbner basis, it is useful to consider the converse of the above insight. We define the ideal of a given complex variety as

\[
I(V) = \{ f \in \mathbb{C}[x] : f(x) = 0 \text{ for all } x \in V \}.
\]

Naively, one could think that for any ideal \( I \subseteq \mathbb{C}[x] \) we have \( I(V(I)) = I \). Obviously the example \( I = \langle x^2 \rangle \) shows that this is incorrect since \( V(I) = \{0\} \) and \( I(\{0\}) = \langle x \rangle \neq \langle x^2 \rangle \). It is useful to find a condition on a system of equations that rules out the possibility
that there is a simpler system that has the same zero set but does not generate the same ideal. After all the whole idea behind the use of Gröbner bases for solving a system of polynomial equations is that the Gröbner basis leads to a very simple system of equations that has the same solution set as the original one. Since by construction it must generate the same ideal, we better want the original ideal to satisfy $I(V(I)) = I$.

For a polynomial $f$ define $f^1 = f$ and $f^n = f^{n-1} f$. For an ideal $I$ the radical of $I$ is defined as

$$\sqrt{I} = \{ f \in \mathbb{K}[x] : \exists m \geq 1 \text{ such that } f^m \in I \}.$$  

The radical $\sqrt{I}$ is itself an ideal and contains $I$. We call an ideal $I$ radical if $I = \sqrt{I}$. A famous theorem, Hilbert’s Nullstellensatz, is the main reason for us to focus on zero-dimensional radical ideals.

**Theorem 2.2** (Hilbert’s Nullstellensatz). If $\mathbb{K} = \mathbb{C}$, then for any ideal $I \subset \mathbb{C}[x]$ we have $I(V(I)) = \sqrt{I}$.

So, each radical ideal corresponds to a complex variety and vice versa. This fact is not true for the computationally convenient case of $\mathbb{K} = \mathbb{Q}$, but we see below that this fact is of no consequence for our analysis. Note that if a given ideal is not radical, then it has a radical which is an ideal in $\mathbb{K}[x]$. In order to identify the complex variety of an ideal it suffices to identify the complex variety of its radical.

### 2.1.1 A Formal Definition of Gröbner Bases

In rather intuitive terms, the general idea behind using Gröbner bases to solve systems of polynomial equations is to view the set of polynomials as a vector space where each component of a vector represents a monomial and the scalars are complex numbers. Every vector has an infinite number of elements but only finitely many of them are nonzero. Thus this vector space is infinite-dimensional. Additionally there is a multiplication defined between two vectors, which is just the multiplication of two polynomials. The Gröbner basis algorithm performs a “kind of” Gaussian elimination to compute a triangular system, which we can then solve. Before we turn to Gröbner bases it is, therefore, helpful to review a feature of Gaussian elimination.

Recall that a matrix is said to be in **echelon form** if it has the shape resulting from a Gaussian elimination. Row echelon form can be obtained by Gaussian elimination, operating on the rows, and column echelon form is computed by column operations.

**Example 2.3** (Gaussian elimination and row echelon form). We provide this short example to recall this basic notion from linear algebra. Consider the following system of linear equations.

\[
\begin{align*}
x + 3y - 2z &= 1, \\
4y + z &= 0, \\
2x + 2y - 5z &= 2.
\end{align*}
\]
In the first step the Gauss algorithm chooses a pivot element, here we choose $x$ from the first equation, and then we eliminate all other occurrences of $x$ in the other equations. So by subtracting two times the first equation from the last one we obtain

$$x + 3y - 2z = 1,$$
$$4y + z = 0,$$
$$-4y - z = 0.$$

As a second pivot we choose $4y$ in the second equation and compute

$$x - \frac{11}{4}z = 1,$$
$$y + \frac{1}{4}z = 0.$$

The system is now in the reduced row echelon form. Thus it is easy to compute the set of solutions as a function of the last variable $z$. (If the system has full rank, then we can simply read of the unique solution for the last variable on the right-hand side and then quickly compute the values of all other variables via backward substitution.) Any linear system of equations can be transformed into the row echelon form by Gaussian elimination.

To extend the idea of Gaussian elimination to polynomials, we need to fix the ordering of the monomial components. For this purpose, we first define the set of all monomials in $n$ variables,

$$\text{Mon}(x_1, \ldots, x_n) = \text{Mon}_n := \{x^\alpha | \alpha \in \mathbb{Z}_+^n\}.$$ We want to introduce an order on this set which is preserved if we multiply two elements by the same monomial.

**Definition 2.4.** A monomial ordering is a total ordering $>$ on the set $\text{Mon}_n$ of all monomials in $n$ variables, which satisfies the following condition:

$$x^\alpha > x^\beta \implies x^\alpha x^\gamma > x^\beta x^\gamma \text{ for all } x^\gamma \in \text{Mon}_n.$$ If $x^\alpha > 1$ for all $x^\alpha \in \text{Mon}_n \setminus \{1\}$ then we call $>$ a global monomial ordering.

In this article, we often use the so-called lexicographical ordering which we denote by $>_lp$ (since the software package SINGULAR uses the notation $lp$) and often simply call the lex order. Given two monomials in $n$ variables, we say that a monomial $x^\alpha$ is greater in this ordering than $x^\beta$ if $\alpha_1 > \beta_1$ or if $\alpha_i = \beta_i$ for $i = 1, \ldots, m$ and $\alpha_{m+1} > \beta_{m+1}$, i.e., if $\alpha$ is greater than $\beta$ lexicographically. We write

$$x^\alpha >_lp x^\beta \iff \alpha > \beta \text{ lexicographically.}$$

Another frequently used example of an ordering is the so-called degree reverse lexicographic ordering denoted by $dp$. We have

$$x^\alpha >_dp x^\beta \iff \deg(x^\alpha) > \deg(x^\beta).$$
or \((\deg(x^\alpha) = \deg(x^\beta))\) and \(\exists 1 \leq i \leq n : \alpha_n = \beta_n, \ldots, \alpha_{i+1} = \beta_{i+1}, \alpha_i < \beta_i\).

The general theory of Gröbner bases can be developed for any given monomial ordering. However, the reader should keep in mind that in order to use Gröbner bases to solve polynomial equations one often needs to focus on the lexicographic ordering.

Now that we have an ordering of the monomials, we can define the largest monomial of a given polynomial, we can identify its coefficient, and we can talk about the leading term of a polynomial being the monomial weighted by its coefficient.

**Definition 2.5.** Let > be a global monomial ordering on \(\text{Mon}(x_1, \ldots, x_n)\) and let

\[ p = a_{\alpha_1}x^{\alpha_1} + a_{\alpha_2}x^{\alpha_2} + \cdots + a_{\alpha_m}x^{\alpha_m} \quad \text{with} \quad x^{\alpha_1} > x^{\alpha_2} > \cdots > x^{\alpha_m} \]

be a polynomial with \(\alpha^m \in \mathbb{Z}_+^n\) for all \(m\) and \(a_{\alpha_1} \neq 0\). Note here that, since every monomial ordering is a total ordering, every polynomial can be arranged in this manner. Then we define the following expressions:

(i) \(\text{LM}(p) := x^{\alpha_1}\) the leading monomial of \(p\);
(ii) \(\text{LC}(p) := a_{\alpha_1}\) the leading coefficient of \(p\);
(iii) \(\text{LT}(p) := a_{\alpha_1}x^{\alpha_1}\) the leading term of \(p\).

We want to argue that in some sense a Gröbner basis to a polynomial system of equations can be compared to the result of Gaussian elimination to a linear system. In order to understand the analog of Gaussian elimination for a system of polynomials, we need to introduce division of monomials. We say a monomial \(x^\alpha\) divides another monomial \(x^\beta\), if \(\alpha_i \leq \beta_i\) for all \(i\), and denote this by \(x^\alpha | x^\beta\). In particular, \(x^\beta = x^\gamma x^\alpha\) for some \(\gamma \in \mathbb{Z}_+^n\). If \(x^\alpha\) does not divide \(x^\beta\) we write \(x^\alpha \not{|} x^\beta\).

Now we can formally define a Gröbner basis.

**Definition 2.6.** Let \(I \subset \mathbb{Q}[x]\) be an ideal generated by the polynomials \(f_1, \ldots, f_k\) and let > be a global monomial ordering on \(\text{Mon}_n\). Furthermore let \(G = \{g_1, \ldots, g_s\} \subset I\). Then \(G\) is called a Gröbner basis of \(I\) with respect to >, if and only if for all \(f \in I\) there exists a \(g \in G\) such that \(\text{LM}(g) | \text{LM}(f)\).

Gröbner bases are not unique and so we may want to choose among them a basis that is particularly useful for solving systems of polynomial equations. For this purpose, the next definition generalizes the concepts “row echelon form” and “reduced row echelon” form to polynomials.

**Definition 2.7.** Let \(G \subset \mathbb{K}[x]\) be any subset of polynomials in the variables \(x = (x_1, \ldots, x_n)\) with coefficients in a field \(\mathbb{K}\).

(1) \(G\) is called interreduced, if \(0 \notin G\) and \(\text{LM}(h) \not{|} \text{LM}(g)\) for all \(h \neq g\) with \(h, g \in G\).
(2) A polynomial \(f = \sum_\alpha a_\alpha x^\alpha\) is called (completely) reduced with respect to \(G\), if for all \(\alpha\) with \(a_\alpha \neq 0\) and for all \(g \in G\) we have that \(\text{LM}(g) \not{|} x^\alpha\).
If $G$ is interreduced, all $g \in G$ are reduced with respect to $G \setminus \{g\}$ and $\text{LC}(g) = 1$ for all $g \in G$, then we call $G$ reduced.

**Example 2.8.** To see how a reduced set of polynomials relates to the more specific notion of the row echelon form we consider a system of linear equations. Let $G = \{x_1 + x_2, -x_2 + x_3\} \subset \mathbb{Q}[x_1, x_2, x_3]$ and order monomials according to the lexicographic ordering, with $x_1 > x_2 > x_3$. We can write the linear system as

$$
\begin{align*}
  x_1 + x_2 &= 0, \\
  -x_2 + x_3 &= 0.
\end{align*}
$$

It is easy to verify that $G$ is interreduced but it is not reduced since $\text{LM}(-x_2 + x_3) = x_2$ and trivially $x_2 | x_2$. In the language of linear algebra this just means that we have a row echelon form but not a reduced one, if the pivot for the first row is $x_1$ and for the second $-x_2$. However, if we reorder the variables as $x_1 > x_3 > x_2$ and impose the corresponding lexicographic order, then $G$ is reduced.

$$
\begin{align*}
  x_1 + x_2 &= 0, \\
  x_3 - x_2 &= 0.
\end{align*}
$$

In other words, if we take $x_1$ as the pivot of the first row and $x_3$ as the pivot of the second row, then we have the reduced row echelon form.

We call a Gröbner basis $G$ reduced if $G$ is a reduced set.

**Theorem 2.9.** Let $I$ be an ideal and $>$ a monomial ordering. Let $G$ be a reduced Gröbner basis of $I$. Then $G$ is unique.

At this point, it is far from obvious why a Gröbner basis has any properties that are useful to determine all solutions. The so-called elimination theorem highlights the connection between Gröbner bases and Gaussian elimination.

### 2.1.2 Elimination Ideals and the Shape Lemma

Given $I = \langle f_1, \ldots, f_s \rangle \subset \mathbb{K}[x_1, \ldots, x_n]$, define the $l$th elimination ideal $I_l$ as the ideal in $\mathbb{K}[x_{l+1}, \ldots, x_n]$ defined by $I_l = I \cap \mathbb{K}[x_{l+1}, \ldots, x_n]$. In other words $I_l$ consists of all “consequences” of $f_1 = \cdots = f_s = 0$ which eliminate the variables $x_1, \ldots, x_l$. Each $I_l$ is an ideal, i.e., there exist polynomials $f_1, \ldots, f_r \in \mathbb{K}[x_{l+1}, \ldots, x_n]$ that generate $I_l$. If $I$ is radical and zero-dimensional the $(n-1)$th elimination ideal must describe the $x_n$-coordinate of all possible solutions to the original system solutions. Since there are finitely many solutions, there must be a univariate polynomial that generates this (at least, if we take $\mathbb{K} = \mathbb{C}$, this is just the product of all $(x_n - b_i)$ terms for all zeros $b_i$). By the Nullstellensatz, this polynomial must itself belong to the ideal (since adding it to the ideal does not change the solution set).

Given a set of polynomials $G = \{f_1, \ldots, f_r\}$, we can obviously define $G \cap \mathbb{K}[x_{l+1}, \ldots, x_n]$ as those polynomials in $G$ which only involve $x_{l+1}, \ldots, x_n$. For general polynomials
and $l > 0$ this set will generally be empty. However, not for Gröbner bases, as the following theorem shows.

**Theorem 2.10 (Elimination Theorem).** Let $I \subseteq \mathbb{K}[x_1, \ldots, x_n]$ be an ideal and let $G$ be a Gröbner basis of $I$ with respect to the lexicographic ordering where $x_1 > x_2 > \cdots x_n$. Then for every $0 \leq l \leq n$, $G_l = G \cap \mathbb{K}[x_{l+1}, \ldots, x_n]$ is a Gröbner basis of the $l$th elimination ideal.

This result now leads us to the Shape Lemma.

**Lemma 2.11 (Shape Lemma).** Let $I \subseteq \mathbb{Q}[x]$ be a radical ideal generated by polynomials $f_1, \ldots, f_n$, such that there are exactly $d$ isolated roots. Further let all $d$ complex roots have a distinct $x_n$-coordinate. If $G$ is a reduced Gröbner basis of $I$ with respect to the lexicographic ordering where $x_1 > \cdots > x_n$, then it has the following form,

$$G = \{x_1 - q_1(x_n), \ldots, x_{n-1} - q_{n-1}(x_n), r(x_n)\},$$

where $r$ is a polynomial of degree $d$ and the $q_i$ have a degree of at most $d - 1$.

Given a zero-dimensional radical ideal $I$, with $V(I) = \{b^{(i)}, \ldots, b^{(d)}\}$ for $b^{(i)} \in \mathbb{C}^n$, for all $i$, the above theorem implies that the reduced lexicographic Gröbner basis of $I$ must contain the univariate polynomial $\Pi_{i=1}^{d} (x_n - b^{(i)}_n)$, which must be a polynomial over $\mathbb{K}$. If across all $i = 1, \ldots, d$, the $b^{(i)}$ are distinct, for each $l = 1, \ldots, n-1$, $b^{(i)}_l$ must be the unique solution to a polynomial involving only $x_l, \ldots, x_n$ with $x_{l+1} = b^{(i)}_{l+1}, \ldots, x_n = b^{(i)}_n$. But for $\mathbb{K} = \mathbb{C}$ this implies that this polynomial must be linear in $x_l$ (otherwise it has more than one solution, the ideal being radical rules out multiple zeros). The Shape Lemma then simply follows by substituting recursively for each $x_l, l = n-1, n-2, \ldots, 1$.

Observe that if the Shape Lemma holds, finding all solutions to a polynomial system of equations, reduces to finding all solutions to a single equation, a task for which there exist efficient numerical methods. Before we turn our attention to this task in Section 2.2, we first want to discuss some basic computational issues.

### 2.1.3 Buchberger’s Algorithm

The basic idea to compute a Gröbner basis is to combine polynomial division and Gaussian elimination. However, unlike with Gaussian elimination, the number of equations may increase during the computations. Furthermore, the choice of which polynomials to use for the elimination steps is much more crucial and difficult. There are now a variety of methods to compute Gröbner bases. The original algorithm by Buchberger implies a constructive existence proof for Gröbner bases and allows us to derive some important properties. Therefore we briefly outline the algorithm in this section.

Given a monomial ordering $>$, we can define for any polynomial $f$ its *multidegree* as follows,

$$md(f) = \max_{> \in Z^n_+} \{\alpha \in Z^n_+ : a_\alpha \neq 0\},$$

where $\max_{>}$ indicates maximization with respect to the monomial ordering $>$. We generalize the division of two polynomials in one variable to multivariate polynomials.
Definition 2.12. Let $>$ be a global monomial ordering on $\text{Mon}_n$. Given any polynomials $f, f_1, \ldots, f_i \in \mathbb{K}[x]$, with $md(f_i) \geq md(f_{i+1})$, and the representation

$$f = a_1 f_1 + \cdots + a_i f_i + r,$$

where $a_i, r \in \mathbb{K}[x]$, and either $r = 0$ or $r$ is a linear combinations of monomials, none of which is divisible by any $\text{LT}(f_1), \ldots, \text{LT}(f_i)$. Furthermore if $a_i f_i \neq 0$ we must have that $md(f) \geq md(a_i f_i)$. The term $r$ is called the remainder of $f$ on division by $f_1, \ldots, f_i$.

The above representation always exists and is unique up to an element of $\mathbb{K} \{0\}$. A formal algorithm which computes $a_1, \ldots, a_i$ and $r$ given any polynomials $f, f_1, \ldots, f_i$ is described in Cox et al. (2007, Chapter 2.3). The algorithm is exact if $\mathbb{K} = \mathbb{Q}$. While the algebra behind the division algorithm is very simple, the algorithm plays a crucial role in the computation of Gröbner basis.

To outline Buchberger’s algorithm for the computation of a Gröbner basis, we need to define an S-polynomial. For this, let $f, g \in \mathbb{K}[x_1, \ldots, x_n]$ with $md(f) = \alpha$ and $md(g) = \beta$. Define $\gamma$ by $\gamma_i = \max\{\alpha_i, \beta_i\}, i = 1, \ldots, n$, and define

$$S(f, g) = \frac{x^\gamma - \alpha}{\text{LT}(f)} f - \frac{x^\gamma - \beta}{\text{LT}(g)} g.$$

Note that $S(f, g)$ is well defined since the two fractions in its definition are both monomials due to the definition of $\gamma$. (The fractions indicate polynomial division.) The S-polynomial is interesting because of the following result.

Theorem 2.13. $G$ is a Gröbner basis if and only if for each $g_i, g_j \in G$, the remainder of $S(g_i, g_j)$ on division by $G$ is zero.

It is easy to see that if $G$ is a Gröbner basis the remainder on division must be zero: Since $S(g_i, g_j) \in \langle G \rangle$ the remainder $r$ of the division is as well. Thus by the Gröbner basis property, there is a $g_k$ such that $\text{LM}(g_k) | r$. This is only possible if $r = 0$; otherwise we could further divide by $g_k$. For a proof of the converse, see Cox et al. (2007, Chapter 2.6).

This result provides the foundation for the proof that the following algorithm always produces a Gröbner basis in finitely many steps. Let $F = \{f_1, \ldots, f_k\}$ be a basis for the ideal $I$. Given a set $F$, we construct a set $G$ which is a Gröbner basis.

Algorithm 2.14 (Buchberger’s Algorithm).

2. $G' := G$.
3. For each pair $p, q \in G'$, $p \neq q$:
   Let $S$ denote the remainder of $S(f, g)$ on division by $G'$; if $S \neq 0$ then $G := G \cup \{S\}$.
4. If $G \neq G'$ go to step 2.
To prove that the algorithm works, first we show that in each iteration \( \langle G \rangle \), i.e., the ideal generated by all polynomials in the finite set \( G \) is a subset of \( I \). If the algorithm terminates, the resulting \( G \) must be a Gröbner basis by the above theorem. It is a bit more involved to show that the algorithm actually does terminate: in each iteration, we must have \( \langle \text{LT}(G') \rangle \subset \langle \text{LT}(G) \rangle \) since \( G' \subset G \). If \( G' \neq G \), the inclusion is strict. The following lemma (called ascending chain lemma) then implies that eventually the inclusion cannot be strict, \( G' = G \) and the algorithm must stop after a finite number of iterations.

**Lemma 2.15 (Ascending Chains).** Let \( I_1 \subset I_2 \subset \ldots \) be an ascending chain of ideals in \( \mathbb{K}[x_1, \ldots, x_n] \). Then there exists a \( N \geq 1 \) such that \( I_N = I_{N+1} = I_{N+2} = \cdots \).

To prove the lemma, consider the set \( I = \bigcup_{i=1}^{\infty} I_i \). The set \( I \) is an ideal. By Hilbert’s Basis Theorem, Theorem 2.1, the ideal \( I \) must be finitely generated, i.e., there must exist \( f_1, \ldots, f_k \) such that \( I = \langle f_1, \ldots, f_k \rangle \), but each of the generators must be contained in some of the \( I_j \), then take \( n \) to be the maximum of these subscripts \( j \).

Note that while this algorithm is well defined independently of the field \( \mathbb{K} \), it can be performed exactly, that is, without numerical error, over \( \mathbb{Q} \).

### 2.1.4 Computing Gröbner Bases with Computer Algebra Systems

For the symbolic computations we employ the computer algebra system **SINGULAR**, see Decker et al. (2012). We also explain how the computations can be performed in **Mathematica**. We provide a series of examples and illustrate how to compute Gröbner bases in numerical practice and highlight some important properties.

**Example 2.16.** To see that the Gröbner basis algorithm can be viewed as a generalization of Gaussian elimination, we first consider a system of linear equations.

\[
\begin{align*}
 x_1 + x_2 + x_3 &= 5, \\
 2x_1 - x_2 + x_3 &= 8, \\
 -x_1 + 2x_2 + 3x_3 &= -1.
\end{align*}
\]

For the computation of a reduced Gröbner basis in **SINGULAR** with respect to the lexicographic ordering where \( x_1 > x_2 > x_3 \), we first need to declare the polynomial ring \( R = \mathbb{Q}[x_1, x_2, x_3] \) with lexicographic ordering. This declaration is achieved via the following command.

\[
\text{ring } R = 0,(x(1),x(2),x(3)),lp;
\]

The expression \( R = 0 \) shows that we work on the field of rational numbers. Next, \( (x(1),x(2),x(3)) \) shows that we consider polynomials in the three variables with the order \( x_1 > x_2 > x_3 \). Alternatively, we could also write \( (x(1..3)) \). Finally, \( lp \) indicates the lexicographic ordering.

We then need to define the ideal which consists of the original system of equations that we have to solve. In this simple linear example, we have
ideal \( \mathbf{I} = x(1) + x(2) + x(3) - 5, 2x(1) - x(2) + x(3) - 8, \\
- x(1) + 2x(2) + 3x(3) + 1; \)

In order to obtain the reduced Gröbner basis, we use the following command.

\[
\text{option("redSB");}
\]

Finally, we compute the Gröbner basis:

\[
\text{groebner(\mathbf{I});}
\]

\[
_1 = 3x(3) - 2 \\
_2 = 9x(2) - 4 \\
_3 = 9x(1) - 35
\]

We would have obtained the same result by Gaussian elimination; the solution to the linear system of equations is \( x_1 = \frac{35}{9}, x_2 = \frac{4}{9}, x_3 = \frac{2}{3} \).

In \textsc{Mathematica} we perform the computation as follows. To set the monomial ordering to be lexicographic, we have to use the following option.

\[
\text{opt} = \{\text{MonomialOrder} -> \text{Lexicographic}\};
\]

Then we declare the variables and compute the Gröbner basis.

\[
\text{vars} = \text{Table}[x[i], \{i, 1, 3\}]
\]

\[
\]

As explained above, for linear systems the reduced Gröbner basis is nothing else but the reduced echelon form. We now turn to the much more interesting case of nonlinear systems.

**Example 2.17.** As a second example, consider the following three nonlinear equations in the three unknowns \( x, y, z \),

\[
x - yz^3 - 2z^3 + 1 = -x + yz - 3z + 4 = x + yz^9 = 0.
\]

The polynomials on the left-hand sides of these equations define a polynomial ideal. We employ \textsc{Singular} to determine a Gröbner basis. This objective is achieved through the following commands.

\[
\text{ring R=0,(x,y,z),lp;}
\]

\[
\text{ideal I=}(x-yz^3-2z^3+1,-x+yz-3z+4,x+yz^9));
\]

\[
\text{ideal G=groebner(I);}\]

To see the output, type at the \textsc{Singular} command line:

\[
\text{G;}
\]

\[
\text{G[1]=}2z11+3z9-5z8+5z3-4z2-1
\]

\[
\text{G[2]=}2y+1yz10+25z8-4yz7-5z6+5z5-5z4+5z3+40z2-31z-6
\]

\[
\text{G[3]=}2x-2z9-5z7+5z6-5z5+5z4-5z3+5z2+1
\]
The expression $2z_{11}$ denotes the term $2 \cdot z^{11}$. We observe that the Shape Lemma holds. **Singular** reverses the order of the polynomials in the Shape Lemma. So, $G[1]$ denotes the univariate polynomial in the last variable, here $z$; next $G[2]$ denotes the second-to-last polynomial that is linear in the second-to-last variable, here $y$; and finally $G[3]$ is the first polynomial that is linear in the first variable, here $x$.

In **Mathematica** the necessary commands are as follows.

```mathematica
vars = {x, y, z};
ideal = {x - y*z^3 - 2*z^3 + 1, -x + y*z - 3*z + 4, x + y*z^9};
gb = GroebnerBasis[Thread[ideal==0], vars];
```

We can also use **Singular** to compute a Gröbner basis for other polynomial orderings. Continuing our example, the following command computes the Gröbner basis under the degree reverse lexicographic ordering, which **Singular** denotes by dp.

```mathematica
ring R=0,(x,y,z),dp;
ideal I=(x-y*z^3-2*z^3+1,-x+y*z-3*z+4,x+y*z^9);
groebner(I);

_[1]=yz-x-3z+4
_[2]=2x^2+2xy+17xz+35z^2-5x+2y-33z-7
_[3]=xz^2+5z^3-4z^2-x-1
_[4]=1000z^4+36xy^2-4y^3-2000z^3-316xy+32y^2-2171xz-5355z^2
    +953x-684y+5829z+701
_[5]=2y^4-58xy^2-3y^3+2250z^3+383xy+y^2+2327xz+7385z^2
    -1600x+1894y-10673z+1705
_[6]=2xy^3+15x^2y-5xy^2+y^3+250z^3+16xy-11y^2+117x+114y-125
```

We observe that the Shape Lemma does not hold for the degree reverse lexicographic ordering. Obviously this Gröbner basis is much less useful for the computation of all solutions to the system of polynomial equations.

We complete this theoretical section by illustrating that the ordering of the variables plays an important role for Gröbner bases. Thus, if we change the ordering and compute the basis anew, we typically obtain a completely new basis as the following example shows.

**Example 2.18.** We consider the following example,

\[
x^4 - y^2 + y - 2 = 0 \\
y^3 - y = 0
\]

with the lexicographic ordering where $x > y$.

```mathematica
ring R = (0),(x,y),lp;
ideal I = x^4-y^2+y-2,y^3-y;
groebner(I);
```
Now we compute the Groebner basis with the lexicographic ordering where $y > x$.

```
ring R = 0, (y, x), lp;
ideal I = x^4-y^2+y-2, y^3-y;
groebner(I);
_[1]=x^8-6x^4+8
_[2]=yx^4-2y+x^4-2
_[3]=y^2-y-x^4+2
```

The two bases of the same ideal are clearly not identical. They even have a different number of elements.

### 2.2 Using Gröbner Bases to Solve Polynomial Systems

Suppose a square system of equations, $f(x) = 0$ with $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and each $f_i$ being a polynomial with rational coefficients, $f_i \in \mathbb{Q}[x_1, \ldots, x_n]$ is given. Then the Shape Lemma, see Lemma 2.11, provides sufficient conditions for the existence of an alternative system $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ with $g_1, \ldots, g_n$ being polynomials with rational coefficients such that for any $x \in \mathbb{R}^n$,

$$f(x) = 0 \Leftrightarrow g(x) = 0;$$

in addition, the last polynomial $g_n(\cdot)$ is only a function of $x_n$, and for all $i = 1, \ldots, n-1$, the polynomial $g_i$ is only a function of $x_n$ and $x_i$ and linear in $x_i$. If the Shape Lemma holds, the computation of all solutions to a polynomial system reduces to finding all roots of univariate polynomials. If the original polynomial system had rational coefficients, then the Gröbner basis will have rational coefficients as well which can be determined without numerical errors. Numerical errors now only occur while computing all roots of the last polynomial $g_n(\cdot)$. And so, for the purpose of finding economic equilibria, we can focus on the numerical approximation of all real roots of a univariate polynomial.\(^5\)

#### 2.2.1 Root Count for Univariate Polynomials

The Fundamental Theorem of Algebra, see Section 4.1.1, states that a univariate polynomial, $f(x) = \sum_{i=0}^{d} a_i x^i$, with rational, real, or complex coefficients $a_i, i = 0, 1, \ldots, d$, has $d$ zeros, counting multiple roots, in the field $\mathbb{C}$ of complex numbers. Importantly for our economic analysis, bounds smaller than $d$ are often available for the number of real zeros. For a finite sequence $a_0, \ldots, a_k$ of real numbers the number of sign changes is the number of products $a_i a_{i+1} < 0$, where $a_i \neq 0$ and $a_{i+1}$ is the next non-zero element of the sequence.

\(^5\) General algebraic solutions only exist for univariate polynomials of up to degree four. The Abel-Ruffini Theorem (Abel’s Impossibility Theorem) states that no such solutions exist for the general case of polynomials of degree five or higher; see Cox et al. (1998).
sequence. Zero elements are ignored in the calculation of the number of sign changes. The classical Descartes’s Rule of Signs, see Sturmfels (2002), states that the number of positive real zeros of \( f \) does not exceed the number of sign changes in the sequence of the coefficients of \( f \). This bound is remarkable because it bounds the number of positive real zeros. It is possible that a polynomial system is of very high degree and has many solutions but the Descartes bound on the number of positive real zeros of the univariate polynomial \( g_n \) in the Shape Lemma proves that the system has a single real positive solution.

The Descartes bound is not tight and often overstates the true number of positive real solutions for many polynomials. Sturm’s Theorem, see Sturmfels (2002) or Bochnak et al. (1998), yields an exact bound on the number of positive real solutions of a univariate polynomial. For a univariate polynomial \( f \), the Sturm sequence of \( f(x) \) is a sequence of polynomials \( f_0, \ldots, f_k \) defined as follows,

\[
f_0 = f, \quad f_1 = f', \quad f_i = f_{i-1} q_i - f_{i-2} \quad \text{for } 2 \leq i \leq k
\]

where \( f' \) denotes the first derivative of \( f \) and \( f_i \) is the negative of the remainder on division of \( f_{i-2} \) by \( f_{i-1} \), so \( q_i \) is a polynomial and the degree of \( f_i \) is less than the degree of \( f_{i-1} \). The sequence stops with the last nonzero remainder \( f_k \). Sturm’s Theorem provides an exact root count; see, e.g., Bochnak et al. (1998) for a proof.

**Lemma 2.19** (Sturm’s Theorem). Let \( f \) be a polynomial with Sturm sequence \( f_0, \ldots, f_k \) and let \( a < b \in \mathbb{R} \) with neither \( a \) nor \( b \) a root of \( f \). Then the number of roots of \( f \) in the interval \( [a, b] \) is equal to the number of sign changes of \( f_0(a), \ldots, f_k(a) \) minus the number of sign changes of \( f_0(b), \ldots, f_k(b) \).

### 2.2.2 Sufficient Conditions for the Shape Lemma

If the Shape Lemma holds, the computation of a Gröbner basis gives directly all solutions to the polynomial system. Let us recall the three sufficient conditions stated in the lemma. We need that (i) the ideal has finitely many complex solutions, (ii) the ideal is radical, and (iii) all roots have distinct \( x_n \)-coordinates. While conditions (i) and (iii) are intuitive, condition (ii) is rather abstract. In practice, the condition can essentially be interpreted as all finitely many zeros having multiplicity one. A simple sufficient condition for this requirement is that whenever \( f(x) = 0 \) the matrix of partial derivatives of \( f \), \( D_x f(x) \), has full rank \( n \). In a slight abuse of notation, we call an ideal regular if its variety has finitely many points and if this full-rank condition holds. In economic applications the first two conditions, typically known as “regularity,” hold if we only consider real solutions (competitive and Nash equilibria are “generically” locally unique and finite in number). It is sometimes difficult to verify that one in fact has finitely many complex solutions. In fact, we present an example below where this is not the case. A simple approach to ensure that all solutions are locally isolated and that the ideal is radical is to add the following additional polynomial equation to the original system.

\[
1 - t \det[D_x f(x)] = 0.
\]
There cannot be a solution in $t$ and $x$ which is not locally unique. Of course this condition may eliminate many complex solutions of the original system. The point is, however, that if we know that all economic solutions are locally isolated and indeed regular, the full-rank condition only eliminates solutions that are of no economic interest.

Condition (iii) holds for a wide variety of problems. In case the condition does not hold, we can always add an additional equation

$$y - \sum_{l=1}^{n} \alpha_l x_l = 0.$$ 

For generic $\alpha$ all solutions to $f(x) = 0$ and $y - \sum_{l=1}^{n} \alpha_l x_l = 0$ will have distinct $y$-coordinates. Therefore, the Shape Lemma holds for the larger system with the ordering $x_1 > x_2 > \cdots > x_n > y$.

### 2.2.3 What If the Shape Lemma Does Not Apply?

The Shape Lemma is useful abstractly because it allows us to transform the original system of equations into a triangular setting. The echelon form from linear algebra has the following equivalent in our more general setting.

**Definition 2.20.** A set of polynomials $f_1, \ldots, f_n \in \mathbb{Q}[x_1, \ldots, x_n]$ is called a triangular system if it has the following form,

$$f_1 = x_1^{d_1} + \sum_{j=0}^{d_1} g_{j}^{(1)}(x_2, \ldots, x_n)x_1^j \in \mathbb{Q}[x_1, x_2, \ldots, x_n]$$

$$f_2 = x_2^{d_2} + \sum_{j=0}^{d_2} g_{j}^{(2)}(x_3, \ldots, x_n)x_2^j \in \mathbb{Q}[x_2, \ldots, x_n]$$

$$\vdots$$

$$f_{n-1} = x_{n-1}^{d_{n-1}} + \sum_{j=0}^{d_{n-1}} g_{j}^{(n-1)}(x_n)x_{n-1}^j \in \mathbb{Q}[x_2, \ldots, x_n]$$

$$f_n = \sum_{j=0}^{d_n} a_{j}x_n^j \in \mathbb{Q}[x_n].$$

If we have a system of equations that has this form then it can be solved by first determining the roots of $f_n$ by a numerical method. Next we plug the results for $x_n$ into the polynomial $f_{n-1}$ and find its roots. Doing this repeatedly we can solve our system of polynomials. Note that finding all real roots to a univariate polynomial is rather simple computationally and can be done with arbitrary precision. For the more general case that the Gröbner basis is not a triangular system, we can use the following solution approach.
**Theorem 2.21.** Let $p_1, \ldots, p_r$ be polynomials in $\mathbb{Q}[x_1, \ldots, x_n]$ with only finitely many common zeros. Then the set of zeros $\{p_1 = \cdots = p_r = 0\}$ is the union of the solution sets to finitely many triangular systems.

This triangular decomposition can be found by repeatedly computing reduced Gröbner bases with respect to a lexicographic ordering. We illustrate the triangularization algorithm in the next example.

**Example 2.22.** Consider the following system

```plaintext
ring R = 0,(x,y,z),lp;
ideal I = z^2-2,y^2+2*y-1,(y+z+1)*x+y*z+z+2,x^2+x+y-1;
groebner(I);
_[1]=z^2-2
_[2]=y^2+2*y-1
_[3]=xy+xz+x+y*z+z+2
_[4]=x^2+x+y-1
```

The solution is not in triangular form; see the third polynomial. We observe that the problem lies with variable $x$. Note that $g_3 = (x + z + 1)x + yz + z + 2$, thus the leading term in $x$ has coefficient $(x + z + 1)$. The triangularization algorithm now branches at this point in two cases; the first case is $x + z + 1 \neq 0$, the second is $x + z + 1 = 0$. This case distinction produces the following decomposition.$^6$

```plaintext
LIB"triang.lib";
triangM(groebner(I));
[1]:
_[1]=z^2-2
_[2]=y+z+1
_[3]=x^2+x-z-2
[2]:
_[1]=z^2-2
_[2]=y-z+1
_[3]=x+z
```

*Singular* and *Mathematica* perform this decomposition automatically if you use the respective solve command. There is no external function in *Mathematica* for the triangular decomposition.

### 2.2.4 Finding All Solutions with Computer Algebra Systems

Recall **Example 2.17**.

---

$^6$ The computation for the $\neq 0$ branch usually requires additional Gröbner basis computations. Here we can just read off the result. We forego a detailed description of the procedure.
> G;
G[1]=2z11+3z9-5z8+5z3-4z2-1
G[2]=2y+18z10+25z8-45z7-5z6+5z5-5z4+5z3+40z2-31z-6
G[3]=2x-2z9-5z7+5z6-5z5+5z4-5z3+5z2+1

We can determine the number of positive real solutions via Sturm's theorem within any interval \((a, b]\). For example, for \(a = 0, b = 1000\) we write in SINGULAR:

```
LIB "rootsur.lib";
poly p=2z11+3z9-5z8+5z3-4z2-1;
sturm(p,0,1000);
```

We first load the library rootsur.lib, then define the polynomial \(p\) and use the command \(\text{sturm(poly,a,b)}\) to determine the number of real zeros—in this case it turns out that the polynomial has only one real positive solution. To solve numerically for all complex solutions in SINGULAR, we first need to load a library:

```
LIB"solve.lib";
```

The following command provides all complex solutions.

```
solve(G);
```

Only one of the solutions is real, the other 10 are complex. SINGULAR prints them all.

In MATHEMATICA, you can restrict the output to only real solutions by appending the \text{Reals} option.

```
gb={2z11+3z9-5z8+5z3-4z2-1,2y+18z10+25z8-45z7-5z6+5z5
-5z4+5z3+40z2-31z-6,
    2x-2z9-5z7+5z6-5z5+5z4-5z3+5z2+1};
Solve[Thread[gb==0],{x,y,z},Reals]
```

### 2.2.5 Parameterized Gröbner Bases

We are also interested in the case where the coefficients of the polynomials contain parameters. Gröbner bases now have a clear advantage compared to the homotopy approach, because we can use them to compute parameterized solutions. A solution here means that we want to find a triangular system in the variables \(x_1, \ldots, x_n\), whose coefficients are polynomials in the parameters \(p_1, \ldots, p_m\), such that for a generic\(^7\) choice of parameters the solutions to the resulting triangular system solve our problem.

Recall that the Gröbner basis algorithm works similar to the Gauss algorithm. Now we do not use the rational numbers as the field we are operating over but we consider the field of rational functions in the parameters. This means that given two polynomials \(g_1, g_2\) in the parameters with \(g_2 \neq 0\) then \(\frac{g_1}{g_2}\) is an element of that field. In other words, we can simply divide by polynomials that only contain parameters. Consider the following example.

---

\(^7\) We mean with “generic” any subset of \(\mathbb{R}^n\), which can be expressed as \(\mathbb{R}^n \setminus S\), where \(S\) is closed and has Lebesgue measure zero.
Example 2.23. We consider a linear system of equations in the variables $x, y$ and the parameter $p$.

\[(p^2 - 1)x + y = 0,\]
\[x + 3y = 1.\]

Now we simply perform the Gauss elimination by subtracting $\frac{1}{p^2 - 1}$ times the first equation from the second.

\[(p^2 - 1)x + y = 0,\]
\[\left(3 - \frac{1}{p^2 - 1}\right)y = 1.\]

Lastly we subtract $\frac{p^2 - 1}{3p^2 - 4}$ times the last equation from the first and normalize the coefficients.

\[x = -\frac{1}{3p^2 - 4},\]
\[y = \frac{p^2 - 1}{3p^2 - 4}.\]

Observe that along the way we divided by $(p^2 - 1)$ and $(3p^2 - 4)$. Thus, the derived solution is not applicable to original systems with $(3p^2 - 4)(p^2 - 1) = 0$.

The example nicely illustrates why Buchberger’s algorithm works for generic parameter values. The error we can incur by simply ignoring the fact that the coefficients are polynomials is represented by a zero set of polynomials in the parameters. These sets are of measure zero.

Every time we divide by a polynomial in the parameters, we implicitly assume that this polynomial is nonzero. Thus, to obtain a solution for all parameter choices we would need to distinguish different cases. Such an approach is called a comprehensive Gröbner basis. (We omit a detailed description here.) All solutions we obtain via the parameterized approach are (only) generically applicable; they do not apply for the roots of a polynomial in the parameters.

It is useful to state a parameterized version of the Shape Lemma, see Kubler and Schmedders (2010a).

Lemma 2.24 (Parameterized Shape Lemma).

Let $E \subset \mathbb{R}^m$ be an open set of parameters, $(x_1, \ldots, x_n) \in \mathbb{C}^n$ a set of variables and let $f_1, \ldots, f_n \in \mathbb{K}[e_1, \ldots, e_m; x_1, \ldots, x_n]$. Assume that for each $\vec{e} = (\vec{e}_1, \ldots, \vec{e}_m) \in E$, the ideal $I(\vec{e}) = (f_1(\vec{e}; \cdot), \ldots, f_n(\vec{e}; \cdot))$ is regular and all complex solutions have distinct $x_n$-coordinates. Then there exist $r, v_1, \ldots, v_{n-1} \in \mathbb{K}[e; y]$ and $\rho_1, \ldots, \rho_{n-1} \in \mathbb{K}[e]$ such that for “almost all” $e \in E$,

\[\{x \in \mathbb{C}^n : f_1(\vec{e}, x) = \cdots = f_n(\vec{e}, x) = 0\} = \{x \in \mathbb{C}^n : \rho_1(\vec{e})x_1 = v_1(\vec{e}; y), \ldots, \rho_n(\vec{e})x_{n-1} = v_{n-1}(\vec{e}; y) \text{ for } r(\vec{e}; x_n) = 0\}.\]
Clearly some of the underlying assumptions are difficult to verify and might not always hold in practice. However, in many economic applications the assumptions do hold and even in applications where they are not satisfied, Gröbner bases can often lead to interesting insights. We illustrate this fact in Section 3 below.

2.2.6 Parameterized Shape Lemma with Computer Algebra Systems
The following variation of Example 2.17, see Kubler and Schmedders (2010b), illustrates how to introduce parameters. Let’s start by leaving the coefficient of the $x$-term in the last equations as a free parameter, that is, the last equation becomes

$$ex + yz^9 = 0.$$ 

In **Singular**, we need to declare the parameters together with the ring in the beginning next to the symbol 0 that indicates that we are working over the field of rational numbers.

```plaintext
ring R=0,e,(x,y,z),lp;
ideal I=(
x-y*z**3-2*z**3+1,
-x+y*z-3*z+4,
e*x+y*z**9);
ideal G=groebner(I);
G;
G[1]=2*z11+3*z9-5*z8+(5e)*z3+(-4e)*z2+(-e)
G[2]=(-e2-e)*y+(-8e-10)*z10+(-10e-15)*z8+(20e+25)*z7
    +(5e)*z6+(-5e)*z5
    +(5e)*z4+(-5e)*z3+(-20e2-20e)*z2+(16e2+15e)*z
    +(3e2+3e)
G[3]=(-e-1)*x+2*z9+5*z7-5*z6+5*z5-5*z4+5*z3-5*z2-1
```

We observe that for all parameters $e > 0$, Descartes’ rule implies that the number of real non-zero solutions for $z$ cannot exceed 3.

In **Mathematica**, we solve for the parameterized Gröbner basis as follows.

```plaintext
polys = {x - y*z^3 - 2*z^3 + 1, -x + y*z - 3*z + 4,
         e*x + y*z^9};
vars = {x, y, z};
GroebnerBasis[polys, vars]
```

**Singular** produces a Gröbner basis for the ideal of parameterized polynomials. Observe that the univariate representation $G[1]$ is a polynomial of degree 11 for any value of $e$. **Figure 1** shows the real roots of the univariate representation for $e \in [-3, 1]$. For positive values of $e$, $G[1]$ has the unique solution $z = 1$. For non-positive values of $e$ there are multiple solutions. Before we decide on the number of real solutions for specific values of $e$, recall that for fixed values of the parameter the parameterized Gröbner basis may not specialize to the correct basis. Here this difficulty becomes obvious. Observe that the leading term of $G[2]$ is $e(-e - 1)y$ and so for $e \in \{-1, 0\}$ the variable $y$ no
longer appears. The same is true for the variable $x$ in $G[3]$ for $e = -1$. Figures 2 and 3 show the real solutions for $G[2]$ and $G[3]$, respectively, for $e \in [-3, 1]$.

As $e \to -1$, the values of $y$ and $x$ grow unbounded in two of the three solutions. Only in one solution their values remain bounded. For $e = -1$ both variables no longer appear in the Gröbner basis. As $e \downarrow 0$ the values of $y$ and $x$ remain bounded in all three solutions.

Instead of using the parameterized basis we need to resolve the original system for $e = 0$ and $e = -1$. For $e = 0$ the resulting Gröbner basis is $\{2z^3 + 3z - 5, y, x + 3z - 4\}$. 
There is a unique real solution, \((1, 0, 1)\). This indicates that as \(e \downarrow 0\) two of the three solutions do not converge to a solution even though all three solutions remain finite. Only the solution with \(z = 1\) converges to a solution of the original system at \(e = 0\). For \(e = -1\) the Gröbner basis is as follows.

\[
\begin{align*}
G[1] &= 2z^9 + 5z^7 - 5z^6 + 5z^5 - 5z^4 + 5z^3 - 5z^2 - 1 \\
G[2] &= 33y + 320z^8 + 10z^7 + 790z^6 - 765z^5 + 740z^4 - 715z^3 + 690z^2 - 665z - 94 \\
G[3] &= 33x + 10z^8 - 10z^7 + 35z^6 - 60z^5 + 85z^4 - 110z^3 + 135z^2 + 5z + 28
\end{align*}
\]

There is a unique real solution, \((-3.37023, -4.63605, 0.965189)\).

## 3. APPLYING GRÖBNER BASES TO ECONOMIC MODELS

We apply Gröbner bases to two simple economic problems. First we consider a static game, and second we examine a standard general equilibrium exchange economy. Both examples are meant to illustrate how typical economic problems can be formulated in terms of systems of polynomials equations and how Gröbner bases can effectively be applied to find all solutions to these equations.

### 3.1 A Bertrand Game

We consider the Bertrand pricing game between two firms from Judd et al. (2012). This example illustrates the various steps that are needed to find all pure-strategy Nash equilibria in a simple game with continuous strategies.
There are two products, \( x \) and \( y \), two firms with firm \( x \) (\( y \)) producing good \( x \) (\( y \)), and three types of customers. Let \( p_x \) (\( p_y \)) be the price of good \( x \) (\( y \)). \( D^1_x, D^2_x, \) and \( D^3_x \) are the demands for product \( x \) by customer type 1, 2, and 3, respectively. Demands \( D^1_y, \) etc., are similarly defined. Type 1 customers only want good \( x \) and have a linear demand curve,
\[
D^1_x = A - p_x; \quad D^1_y = 0.
\]
Type 3 customers only want good \( y \) and have a linear demand curve,
\[
D^3_x = 0; \quad D^3_y = A - p_y.
\]
Type 2 customers demand both commodities. Let \( n \) be the number of type 2 customers. We assume that the two goods are imperfect substitutes for type 2 customers with a constant elasticity of substitution between the two goods and a constant elasticity of demand for a composite good. These assumptions imply the demand functions
\[
D^2_x = np_x^{-\sigma} \left( p_x^{1-\sigma} + p_y^{1-\sigma} \right)^{\frac{\gamma - \sigma}{1+\sigma}},
\]
\[
D^2_y = np_y^{-\sigma} \left( p_x^{1-\sigma} + p_y^{1-\sigma} \right)^{\frac{\gamma - \sigma}{1+\sigma}}.
\]
where \( \sigma \) is the elasticity of substitution between \( x \) and \( y \), and \( \gamma \) is the elasticity of demand for a composite good. Total demand for good \( x \) (\( y \)) is given by
\[
D_x = D^1_x + D^2_x + D^3_x,
\]
\[
D_y = D^1_y + D^2_y + D^3_y.
\]
Let \( m \) be the unit cost of production for each firm. Profit for good \( x \) is \( R_x = (p_x - m)D_x \); \( R_y \) is similarly defined. Let \( MR_x \) be marginal profits for good \( x \); similarly for \( MR_y \). Equilibrium prices satisfy the necessary conditions
\[
R_x = MR_y = 0.
\]
Let
\[
\sigma = 3, \quad \gamma = 2, \quad n = 2700, \quad m = 1, \quad A = 50.
\]
The marginal profit functions are as follows.
\[
R_x = 50 - p_x + (p_x - 1) \left( -1 + \frac{2700}{p_x^6 \left( p_x^{-2} + p_y^{-2} \right)^{3/2}} - \frac{8100}{p_x^4 \sqrt{p_x^{-2} + p_y^{-2}}} \right) + \frac{2700}{p_x^3 \sqrt{p_x^{-2} + p_y^{-2}}},
\]
\[ R_y = 50 - p_y + (p_y - 1) \left( -1 + \frac{2700}{p_y^6(p_x^{-2} + p_y^{-2})^{3/2}} - \frac{8100}{p_y^4(p_x^{-2} + p_y^{-2})} \right) + \frac{2700}{p_y^3\sqrt{p_x^{-2} + p_y^{-2}}}. \]

### 3.1.1 Polynomial Equilibrium Equations

We first construct a polynomial system. The system we construct must contain all the equilibria, but it may have extraneous solutions. The extraneous solutions present no problem because we can easily identify and discard them. Let \( Z \) be the square root term

\[ Z = \sqrt{p_x^{-2} + p_y^{-2}}, \]

which implies

\[ 0 = Z^2 - \left( p_x^{-2} + p_y^{-2} \right). \]

This is not a polynomial. We gather all terms into one fraction and extract the numerator, which is the polynomial we include in our polynomial system to represent the variable \( Z \),

\[ 0 = -p_x^2 - p_y^2 + Z^2 p_x^2 p_y^2. \] (1)

We next use the \( Z \) definition to eliminate radicals in \( MR_x \) and \( MR_y \). Again we gather terms into one fraction and extract the numerator. The other two equations of our polynomial are as follows:

\[ 0 = -2700 + 2700p_x + 8100Z^2 p_x^2 - 5400Z^2 p_x^3 + 51Z^3 p_x^6 - 2Z^3 F^7, \] (2)

\[ 0 = -2700 + 2700p_y + 8100Z^2 p_y^2 - 5400Z^2 p_y^3 + 51Z^3 p_y^6 - 2Z^3 F^7. \] (3)

Any pure-strategy Nash equilibrium is a solution of the polynomial system (1, 2, 3).

### 3.1.2 Solving the System with SINGULAR

To solve this system with SINGULAR we type the following

```plaintext
int n = 2700;
ring R = 0,(px,py,z),lp;
poly f1,f2,f3;
f1 = -(px^2)-py^2+z^2*px^2*py^2;
f2 = -(n)+n*px+3*n*z^2*px^2-2*n*z^2*px^3
    +51*z^3*px^6-2*z^3*px^7;
f3 = -(n)+n*py+3*n*z^2*py^2-2*n*z^2*py^3
    +51*z^3*py^6-2*z^3*py^7;
```
ideal I = f1,f2,f3;
ideal G = groebner(I);
LIB"solve.lib";
solve(G);

We find 62 solutions, of which 44 are complex and 18 are real. Nine of the real solutions contain negative values and are thus of no economic interest. That leaves us with nine candidates for equilibria. Checking the second-order conditions of the firms’ optimization problems eliminates another five solutions. Finally, checking the remaining four solutions for global optimality, we observe that there are two Bertrand equilibria, \( (p_x, p_y) = (2.168, 25.157) \) and \( (p_x, p_y) = (25.157, 2.168) \). For more details, see Judd et al. (2012).

Next we are interested in having \( n \) as a parameter to compute the manifold of solutions.

### 3.1.3 The Manifold of Positive Solutions

We can determine the manifold of solutions by computing a parametric Gröbner basis.

```
ring R = (0,n),(z,px,py),dp;
poly f1,f2,f3;
f1 = -(px^2)-py^2+z^2*px^2*py^2;
f2 = -(n)+n*px+3*n*z^2*px^2-2*n*z^2*px^3
     +51*z^3*px^6-2*z^3*px^7;
f3 = -(n)+n*py+3*n*z^2*py^2-2*n*z^2*py^3
     +51*z^3*py^6-2*z^3*py^7;
ideal I = f1,f2,f3;
ideal G = groebner(I,"hilb");
LIB"teachstd.lib";
ideal rad=radical(G);
ring R2 = (0,n),(z,px,py),lp;
ideal I = fetch(R,rad);
ideal G = groebner(I,"hilb");
```

The Shape Lemma does not hold in this case, since the ideal is not radical. To remedy this problem, we compute the radical of the ideal in a computationally less intensive ordering. Then we transfer this new ideal to a different ring, equipped with the lexicographical ordering. Finally computing the Gröbner basis there leads to a triangular set.

Figure 4 shows the values of \( p_x \) in all positive real solutions as a function of \( n \). For more details on the solutions that are actually equilibria, see Judd et al. (2012).

### 3.2 A Simple Walrasian Exchange Economy

We consider a pure exchange economy with \( H \) agents and \( L \) commodities. Each agent \( h \in \mathcal{H} = \{1, 2, \ldots, H\} \) has CES utility, with marginal utility of the form

\[
v'_h(c) = (\alpha_h^h)^{-\sigma_h}(c_i)^{-\sigma_h}.
\]
This transformation of the standard CES-form may appear unusual at first but considerably simplifies the notation during our analysis. We need to assume that $\sigma_h$ is a rational number for all $h \in \mathcal{H}$ and set $\sigma_h = \frac{N}{M^h}$ such that the greatest common divisor of the natural numbers $N$ and $M^h$ is equal to one for at least one $h \in \mathcal{H}$.

### 3.2.1 Polynomial System and Equilibria

After transforming agents’ first-order conditions into polynomial expressions we obtain the specific equations for our CES-framework.

$$
\alpha_h^N (\epsilon_i^h)^N (\lambda^h)^{M^h} p_i^{M^h} - 1 = 0, \quad h \in \mathcal{H}, \quad l = 1, \ldots, L,
$$

$$
\sum_{l=1}^L p_l (\epsilon_i^h - \epsilon_l^h) = 0, \quad h \in \mathcal{H},
$$

$$
\sum_{h=1}^H \epsilon_i^h - \epsilon_l^h = 0, \quad l = 1, \ldots, L - 1,
$$

$$
\sum_{l=1}^L p_l - 1 = 0.
$$

We can greatly reduce the running times of SINGULAR if we write the equilibrium equations slightly differently. In particular, we normalize $p_1 = 1$ and eliminate all Lagrange multipliers, $\lambda^h = 1/(\alpha_{h1} \epsilon_1^h)^{N/M^h}$. Defining $q_l = p_1^{1/N}, l = 2, \ldots, L$, we obtain a similar system of equations, which has the same real positive solutions but often fewer complex
and negative real solutions.

\[ \alpha_{h1c_1^h} - \alpha_{h1c_1^M} = 0, \quad h \in H, \quad l = 2, \ldots, L, \quad (5) \]

\[ c_1^h - e_1^h + \sum_{l=2}^{L} q_l^N (c_i^h - e_i^h) = 0, \quad h \in H, \quad (6) \]

\[ \sum_{h=1}^{H} c_l^h - e_l^h = 0, \quad l = 1, \ldots, L - 1. \quad (7) \]

The following theorem states properties of the real solutions to this system of equations. The statement is useful for a choice of ordering for the variables to ensure that the Shape Lemma holds.

**Theorem 3.1.** All real solutions \( c^H, q \) to Eqs. (5)–(7) satisfy \( c^h \gg 0 \) whenever \( q \gg 0 \). Moreover, if \( N \) and \( M^h \) are odd for all \( h \in H \), all real solutions satisfy \( q \gg 0 \).

**Proof.** Suppose \( c^H, q \) solve (5)–(7), \( q \gg 0 \) but \( c_i^h < 0 \) for some \( h, l \). Then Eq. (5) implies that \( c_i^h \ll 0 \) for this agent \( h \), but then the budget Eq. (6) cannot hold for this agent.

Now assume \( N, M^h \) odd and \( q_l < 0 \) for at least one \( l \). Define \( \tilde{H} = \{ h : c_i^h > 0 \} \). Market clearing implies that this set is non-empty. Moreover, the budget equations for the agents \( h \in \tilde{H} \) imply

\[ \sum_{h \in \tilde{H}} \left( c_1^h - e_1^h + \sum_{l=2}^{L} q_l^N (c_i^h - e_i^h) \right) = 0. \]

By definition of \( \tilde{H} \), \( \sum_{h \not\in \tilde{H}} (c_1^h - e_1^h) \leq 0 \) and with market clearing \( \sum_{h \in \tilde{H}} (c_1^h - e_1^h) \geq 0 \). By (5), whenever \( q_l < 0 \), then \( c_i^h < 0 \) and therefore \( q_l^N (c_i^h - e_i^h) > 0 \) for all \( h \in \tilde{H} \). Similarly, if \( q_l > 0 \), then \( c_i^h < 0 \) for all \( h \not\in \tilde{H} \). By market clearing \( \sum_{h \in \tilde{H}} (c_i^h - e_i^h) \geq 0 \) and thus \( \sum_{h \in \tilde{H}} q_l^N (c_i^h - e_i^h) \geq 0 \). In total, since by assumption there is at least one \( l \) with \( q_l < 0 \),

\[ \sum_{h \in \tilde{H}} \left( c_1^h - e_1^h + \sum_{l=2}^{L} q_l^N (c_i^h - e_i^h) \right) > 0, \]

yielding a contradiction. Furthermore, the case \( q_l = 0 \) for some \( l \) is ruled out since this implies that \( c_i^h = 0 \) for all \( h \in H \), contradicting market clearing.

### 3.2.2 Finding All Equilibria with SINGULAR

We only show how to perform the computations with SINGULAR; the main steps are similar in MATHEMATICA. We consider simple exchange economies with \( H \geq 2 \) agents and \( L = H \) commodities. Each agent \( h \) is only endowed with commodity \( l = h \), and we assume \( e_i^h = 1 \) and \( e_i^l = 0 \) for all \( h \neq l \). We consider different values for \( H \) and fix the parameter \( \sigma = 3 \). We assume that \( \alpha_i^h = 1 \) and for each \( l \neq h \) we assume \( \alpha_i^l = a > 1 \). In the following computations we vary the parameter \( a \).
Example 3.2. \((H = L = 2)\) We first consider the simple \(2 \times 2\) case. The polynomial system (5) and (6) with market clearing (7) used to substitute out \(c^H_l\) for all \(l\) and writing \(\bar{c}^H_l = c^H_l - c^H_h\), we obtain the following system in SINGULAR.

\[
\begin{align*}
\text{int } n &= 3; \\
\text{ring } R &= (0,a),x(1..n),lp; \\
\text{option(redSB)}; \\
\text{ideal } I &= ( \\
-a(0+x(2))*x(3)+(1+x(1)), \\
-(1-x(2))*x(3)+a(0-x(1)), \\
x(1)+x(2)*x(3)**3); \\
\text{groebner}(I); \\
_1 &= x(3)^3+(-a)*x(3)^2+(a)*x(3)-1 \\
_2 &= (a^2-1)*x(2)+(-a)*x(3)^2+(a^2)*x(3)+(-a^2+1) \\
_3 &= (a^2-1)*x(1)+(a)*x(3)-1
\end{align*}
\]

The first equation has three solutions

\[x_3 = 1, \quad x_3 = \frac{1}{2}\left(-1 + a \pm \sqrt{-3 - 2a + a^2}\right).\]

The last two solutions are complex for \(a < 3\) and so we have a unique equilibrium for \(a < 3\); for \(a = 3\) we are in the non-generic case of a single Walrasian equilibrium of multiplicity 3 since the first polynomial is then \((x_3 - 1)^3\). For all \(a > 3\) the economy has exactly three competitive equilibria.

Example 3.3. \((H = L = 3)\) We extend the previous example to three agents and three goods.

\[
\begin{align*}
\text{int } n &= 8; \\
\text{ring } R &= (0,a),x(1..n),lp; \\
\text{option(redSB)}; \\
\text{ideal } I &= ( \\
-a(0+x(2))*x(7)+(1+x(1)), \\
-a(0+x(3))*x(8)+(1+x(1)), \\
-(1+x(5))*x(7)+a(0+x(4)), \\
-(0+x(6))*x(8)+(0+x(4)), \\
-(0-x(2)-x(5))*x(7)+(0-x(1)-x(4)), \\
-(1-x(3)-x(6))*x(8)+a(0-x(1)-x(4)), \\
x(1)+x(2)*x(7)**3+x(3)*x(8)**3, \\
x(4)+x(5)*x(7)**3+x(6)*x(8)**3); \\
\text{ideal } G &= \text{groebner}(I); \\
\end{align*}
\]

The Shape Lemma does not hold and the resulting Gröbner basis has nine elements. The first three polynomials determine \(x(8)\) and \(x(7)\),
Computing All Solutions to Polynomial Equations in Economics

\[ G[1] = 2x(8)^7 + (-3a+3)x(8)^6 + (a^2-2a+6)x(8)^5 + (-a^2+2a-6)x(8)^4 + (3a-1)x(8)^3 + (-a^2+2a-6)x(8)^2 + (3a-3)x(8) - 2 \]

\[ G[2] = (a^3+3a^2+3a+2)x(7)x(8) + (-a^3-3a^2-3a-2)x(7) + (-2a^2-2a-2)x(8)^6 + (3a^3-6a-9)x(8)^5 + (-a^4+a^3+4a^2+2a-9)x(8)^4 + (-2a^2-8a-8)x(8)^3 + (-2a^3+3a^2+9a+11)x(8)^2 + (a^4-a^3-4a^2-2a+9)x(8) + (-a^3+a^2+7a+8) \]

\[ G[3] = (a^2+a+1)x(7)^3 + (-a^3-a^2-a)x(7)^2 + (a^3+2a^2+2a+1)x(7) + (4a+2)x(8)^6 + (-6a^2+3a+3)x(8)^5 + (2a^3-3a^2+6a+4)x(8)^4 + (-a^2-a-1)x(8)^3 + (-a^3+4a^2-5a-4)x(8)^2 + (-a^3+4a^2-5a-4)x(8) + (-4a-2) \]

The first equation determines \( x(8) \). For \( x(8) \neq 1 \) the second equation uniquely pins down \( x(7) \). However, for \( x(8) = 1 \) the second polynomial of the Gröbner basis does not contain \( x(7) \) and becomes identical to the first with the term \((x-1)\) factored out. In this case the third element determines \( x(7) \). However, this expression is cubic in \( x(7) \) and therefore potentially has three solutions. Given \( x(8) = 1 \) we obtain, in addition to \( x(7) = 1 \), the two additional solutions \( x(7) = (\frac{-1}{2} - a \pm \sqrt{-7 - 2a + a^2}) \). These solutions are real for all \( a \geq 1 + 2\sqrt{2} \). Despite the fact that the first polynomial is of degree seven all of its seven solutions can be obtained in closed form. In addition to the solution \( x(8) = 1 \), the polynomial \( G[1] \) has two solutions that are always complex and has another four solutions that are real for all \( a \geq 1 + 2\sqrt{2} \). We obtain the following real solutions \( x_8 \in \{ \frac{1}{2}(-1 + a - \sqrt{-7 - 2a + a^2}), \frac{1}{2}(-1 + a + \sqrt{-7 - 2a + a^2}), \frac{1}{2}(-1 + a - \sqrt{-7 - 2a + a^2}), \frac{1}{2}(-1 + a + \sqrt{-7 - 2a + a^2}) \} \).

In sum, the system potentially has seven real solutions; in fact, we can verify all of them to be Walrasian equilibria.

**Example 3.4.** \((H = L = 4)\) We fix \( a = 30 \). As before, the Shape Lemma does not hold. Unfortunately in this example the situation is worse since the solution set is not zero-dimensional. The Singular code returns an error message.

```plaintext
int n = 15;
ring R= 0,x(1..n),lp;
option(redSB);
ideal I = (-30*(0+x(2))*x(13)+(1+x(1))),(H = L = 4) We fix a = 30. As before, the Shape Lemma does not hold. Unfortunately in this example the situation is worse since the solution set is not zero-dimensional. The Singular code returns an error message.

```
\[-30*(0+x(3))*x(14)+(1+x(1)),
-30*(0+x(4))*x(15)+(1+x(1)),
-(1+x(6))*x(13)+30*(0+x(5)),
-(0+x(7))*x(14)+(0+x(5)),
-(0+x(8))*x(15)+(0+x(5)),
-(0+x(10))*x(13)+(0+x(9)),
-(1+x(11))*x(14)+30*(0+x(9)),
-(0+x(12))*x(15)+(0+x(9)),
-(0-x(2)-x(6)-x(10))*x(13)+(0-x(1)-x(5)-x(9)),
-(0-x(3)-x(7)-x(11))*x(14)+(0-x(1)-x(5)-x(9)),
-(1-x(4)-x(8)-x(12))*x(15)+30*(0-x(1)-x(5)-x(9)),
\]
\[x(1)+x(2)*x(13)**3+x(3)*x(14)**3+x(4)*x(15)**3,
\[x(5)+x(6)*x(13)**3+x(7)*x(14)**3+x(8)*x(15)**3,
\[x(9)+x(10)*x(13)**3+x(11)*x(14)**3+x(12)*x(15)**3);
\]
\[\text{ideal } G=groebner(I);
\text{LIB"solve.lib";}
\text{solve}(G);
\text{? ideal not zero-dimensional}
\text{? leaving solve.lib::solve}
\]

The error message indicates that the solution set is infinite. Thus we have to reduce the problem to instances we can solve. One way of doing this is to use the so-called primary decomposition.\(^8\) It returns a list of ideals whose intersection is the original ideal. In particular, it will split off any multidimensional components. The computation is usually very costly, but for this example it works.

\[\text{LIB"primdec.lib";}
\text{list lprim=primdecGTZ(G)};
\]

The last eight components are the zero-dimensional ones and we can compute the solutions as we have done before.

\[\text{int } i = 0;
\text{for} (i=2;i<=9;i++) {
\text{ solve(lprim[i][2]);}
};
\]

The first component here is not zero-dimensional. We examine the first polynomial of its Gröbner basis.

\[\text{ideal } G2 = \text{groebner(lprim[1][2]);}
G2[1];
\text{x(14)^2+x(14)*x(15)+x(14)+x(15)^2+x(15)+1}
\]

\(^8\) We omit mathematical details on this advanced concept from commutative algebra.
Note that this polynomial is positive for all values of $x_{14}$ and $x_{15}$. So it has no real solutions and therefore this solution component has no real solutions. Therefore, the solutions of the zero-dimensional components form all equilibria. In total this system of equations has 15 real solutions—all of them are competitive equilibria. Equilibrium prices $q_2, q_3, q_4$, with $q_1 = 1$ are displayed in Table 1.

The three examples illustrate nicely how the number of equilibria naturally increases as the number of agents and goods increases at the same time. Kubler and Schmedders (2010b) show that for identical $\sigma$ the number of equilibria does not increase if only the number of goods increases. What is crucial in this construction is that each agent puts a large weight on exactly one commodity in his or her utility function.

### 4. ALL-SOLUTION HOMOTOPY METHODS

In this section we provide a brief introduction to all-solution homotopy methods for polynomial systems of equations following the paper Judd et al. (2012) and the references therein, most notably Sommese and Wampler (2005).

#### 4.1 Brief Introduction to All-Solutions Homotopies

All-solution homotopy methods rely on results from complex analysis and algebraic geometry. So we begin with some additional definitions and theorems which are necessary for an understanding of the underlying mathematics.
4.1.1 Mathematical Background

We begin with a basic theorem, see Cox et al. (2007).

**Theorem 4.1 (Fundamental Theorem of Algebra).** Let $f$ be a polynomial of degree $d > 0$. Then $f$ has a root in $\mathbb{C}$.

A direct consequence of this theorem is that any univariate polynomial $f$ of degree $d$ over the complex numbers can be written as $f(z) = c(z - b_1)^{r_1} (z - b_2)^{r_2} \cdots (z - b_l)^{r_l}$ with $c \in \mathbb{C} \setminus \{0\}$, $b_1, b_2, \ldots, b_l \in \mathbb{C}$, $r_1, r_2, \ldots, r_l \in \mathbb{N}$, and $\sum_{i=1}^l r_i = d$. The exponent $r_j$ is called the multiplicity of the root $b_j$. For example, the polynomial $z^3$ has the single root $z = 0$ with multiplicity 3. A simple polynomial of degree $d$ with $d$ distinctive complex roots is $g(z) = z^d - 1$, whose roots are $r_k = e^{\frac{2\pi ik}{d}}$ for $k = 0, \ldots, d - 1$. These roots are called the $d$th roots of unity. They become relevant to us below.

We continue with the definition of a homogeneous polynomial.

**Definition 4.2.** A polynomial $f$ over the variables $z_1, \ldots, z_n$ is said to be homogeneous of degree $d$, if for any $a \in \mathbb{C}$

$$f(az_1, \ldots, az_n) = a^d f(z_1, \ldots, z_n).$$

Any polynomial $f$ of degree $d$ can be written as

$$f = \sum_{j=0}^d f^{(j)},$$

where $f^{(j)}$ is a homogeneous polynomial of degree $j$ and $f^{(d)}$ is not the zero polynomial.

Note that we can interpret a polynomial $f$ in the variables $z_1, \ldots, z_n$ as a function of $f : \mathbb{C}^n \to \mathbb{C}$. The following class of functions contains the set of polynomials.

**Definition 4.3.** Let $U \subset \mathbb{C}^n$ be an open subset and $f : U \to \mathbb{C}$ a function. Then we call $f$ analytic at the point $b = (b_1, \ldots, b_n) \in U$ if and only if there exists a neighborhood $V \subseteq U$ of $b$ such that

$$f(z) = \sum_{j=0}^\infty \left( \sum_{d_{1}+\ldots+d_{n}=j} a_{(d_{1},\ldots,d_{n})} \prod_{k=1}^{n} (z_k - b_k)^{d_k} \right), \quad \forall z \in V,$$

where $a_{(d_{1},\ldots,d_{n})} \in \mathbb{C}$, i.e., the above power series converges to the function $f$ on $V$. It is called the series expansion of $f$ at $b$. A function $f$ is called analytic on $U$, if it is analytic at each point of $U$.

Obviously every function given by polynomials is analytic with a globally convergent series expansion. In general, however, $V \subset \not\subseteq U$ and the series expansion are divergent outside of $V$. The following theorem generalizes the Implicit Function Theorem to complex space and analytic functions.
Theorem 4.4 (Implicit Function Theorem). Let
\[ H : \mathbb{C} \times \mathbb{C}^n \rightarrow \mathbb{C}^n \text{ with } (t, z_1, \ldots, z_n) \mapsto H(t, z_1, \ldots, z_n) \]
be an analytic function. Denote by \( D_z H = \left( \frac{\partial H_j}{\partial z_i} \right)_{i,j=1,\ldots,n} \) the submatrix of the Jacobian of \( H \) containing the partial derivatives with respect to \( z_i, i = 1, \ldots, n \). Furthermore let \( (t_0, x_0) \in \mathbb{C} \times \mathbb{C}^n \) such that \( H(t_0, x_0) = 0 \) and \( \det D_z H(t_0, x_0) \neq 0 \). Then there exist neighborhoods \( T \) of \( t_0 \) and \( A \) of \( x_0 \) and an analytic function \( x : T \rightarrow A \) such that \( H(t, x(t)) = 0 \) for all \( t \in T \). Furthermore the chain rule implies that
\[
\frac{\partial x}{\partial t}(t_0) = -D_z H(t_0, x_0)^{-1} \cdot \frac{\partial H}{\partial t}(t_0, x_0).
\]

The basic ingredient for homotopy methods is the path.

Definition 4.5. Let \( A \subset \mathbb{C}^n \) be an open or closed subset. An analytic function \( x : [0, 1] \rightarrow A \) or \( x : [0, 1) \rightarrow A \) is called a path in \( A \).

Definition 4.6. Let \( H(t, z) : \mathbb{C}^{n+1} \rightarrow \mathbb{C}^n \) and \( x : [0, 1] \rightarrow \mathbb{C}^n \) an analytic function such that \( H(t, x(t)) = 0 \) for all \( t \). Then \( x \) defines a path in \( \{ (t, x) \in \mathbb{C}^{n+1} \mid H(t, x) = 0 \} \). We call the path regular, if \( \{ t \in [0, 1) \mid H(t, x(t)) = 0, \det D_z H(t, x(t)) = 0 \} \neq \emptyset \).

The next concept is needed to ensure that our paths connect to all solutions.

Definition 4.7. Let \( A \subset \mathbb{C}^n \). We call \( A \) pathwise connected, if for all points \( a_1, a_2 \in A \) there exists a continuous function \( x : [0, 1] \rightarrow A \) such that \( x(0) = a_1 \) and \( x(1) = a_2 \).

Lastly we need the following notion from topology, which also gives the name to the method.

Definition 4.8. Let \( U, V \subset \mathbb{C}^n \) be open subsets and \( h_0 : U \rightarrow V, h_1 : U \rightarrow V \) be continuous functions. Let
\[ H : [0, 1] \times U \rightarrow V \]
be a continuous function such that \( H(0, z) = h_0(z) \) and \( H(1, z) = h_1(z) \). Then we call \( H \) a homotopy from \( h_0 \) to \( h_1 \).

4.1.2 All Roots of Univariate Polynomials
Homotopy methods have a long history in economics, see Eaves and Schmedders (1999), for finding a single solution to a system of nonlinear equations. The all-solutions homotopy for polynomial systems was first introduced by Garcia and Zangwill (1977) and Drexler (1977). These papers sparked an active field of research that is still advancing.
today. See Sommese and Wampler (2005) for an overview. In this section, following Sommese and Wampler (2005) and the many cited works therein, we provide some intuition for the theoretical foundation.

The basic idea of the homotopy approach is to find an easier way to solve the system of equations and continuously transform it into our target system. Consider the univariate polynomial \( f(z) = \sum_{i \leq d} a_i z^i \) with \( a_d \neq 0 \) and \( \text{deg}(f) = d \). By the Fundamental Theorem of Algebra we know that \( f \) has precisely \( d \) complex roots, counting multiplicities. Now we can define a homotopy \( H \) from \( g \) to \( f \) by setting \( H = (1 - t)(z^d - 1) + tf \). Under the assumption that \( \frac{\partial H}{\partial z}(t, z) \neq 0 \) for all \((t, z)\) satisfying \( H(t, z) = 0 \) and \( t \in [0, 1] \) the Implicit Function Theorem (Theorem 4.4) states that each root \( r_k \) of \( g \) gives rise to a path that is described by an analytical function. The idea is now to start at each solution \( z = r_k \) of \( H(0, z) = 0 \) and to follow the resulting path until a solution \( z \) of \( H(1, z) = 0 \) has been reached. The path-following can be done numerically. As a first step we use Euler’s method, a so-called first-order predictor. We choose a \( \varepsilon > 0 \) and calculate

\[
\tilde{x}_k(0 + \varepsilon) = x_k(0) + \varepsilon \frac{\partial x_k}{\partial t}(0),
\]

where the \( \frac{\partial x_k}{\partial t}(0) \) are implicitly given by Theorem 4.4. Then this first estimate is corrected using Newton’s method with starting point \( \tilde{x}_k(0 + \varepsilon) \). By this we solve \( H(\varepsilon, z) = 0 \) for \( z \) and sets \( x_k(\varepsilon) = z \). This approach is therefore called a predictor-corrector method (see, for example, Allgower and Georg (2003)). Another well-known approach is the Runge-Kutta method.

**Example 4.9.** Consider the polynomial \( f(z) = z^3 + z^2 + z + 1 \). The zeros are \(-1, i, -i\). As a start polynomial we choose \( g(z) = z^2 - 1 \). Then a homotopy from \( g \) to \( f \) can be defined as

\[
H(t, z) = (1 - t)(z^3 - 1) + t(z^3 + z^2 + z + 1).
\]

This homotopy generates the three solution paths shown in Figure 5. The starting points of the three paths, \(-\frac{1}{2} - \frac{\sqrt{3}}{2}i, -\frac{1}{2} + \frac{\sqrt{3}}{2}i, 1\), respectively, are indicated by circles. The respective end points, \(-i, i, \) and \(-1\), are indicated by squares.

This brief description outlines the idea behind any homotopy solution methods. There are two potential pitfalls which we must address. The paths might cross, or they might bend sideways and diverge. For general homotopies, there is no way around these problems. But in the polynomial case we can circumvent them. The following example from Judd et al. (2012) illustrates the problems.

**Example 4.10.** Let \( f(z) = 5 - z^2 \) and \( g(z) = z^2 - 1 \). Then a homotopy from \( g \) to \( f \) can be defined as

\[
H(t, z) = t(5 - z^2) + (1 - t)(z^2 - 1) = (1 - 2t)z^2 + 6t - 1.
\]
Observe that $H(\frac{1}{6}, z) = \frac{2}{3}z^2$ has the double root $z = 0$, so $\det D_zH(\frac{1}{6}, 0) = 0$. These points are non-regular and the corresponding assumption of the Implicit Function Theorem is not satisfied. Non-regular points cause additional trouble for the Newton corrector step in the path-following algorithm.

This homotopy has an additional problem. Since $H(\frac{1}{2}, z) = 2$, which has no zero at all, there can be no solution path from $t = 0$ to $t = 1$. The coefficient of the leading term $(1 - 2t)z^2$ has become 0 and so the degree of the polynomial $H$ drops at $t = \frac{1}{2}$. Figure 6
displays the set of zeros of the homotopy. The two paths starting at $\sqrt{5}$ and $-\sqrt{5}$ diverge as $t \to \frac{1}{2}$.

The idea behind resolving these issues ties into the relationship with generic parameters as seen in Example 2.23. The key difference is that we now also vary the parameters over $\mathbb{C}$. The following theorem illustrates why we are forced to do that.

**Theorem 4.11.** Let $F = (f_1, \ldots, f_k) = 0$ be a system of polynomial equations in $n$ variables, with $f_i \neq 0$ for some $i$. Then $\mathbb{C}^n \setminus \{F = 0\}$ is a pathwise connected and dense subset of $\mathbb{C}^n$.

Let $n = 2$, $k = 1$ and set $f_1(x_1, x_2) = x_1$. If we now only regard the real numbers, $(x_1, x_2) \in \mathbb{R}^2$, then the zero set $\{(x_1, x_2) \in \mathbb{R}^2 : f_1(x_1, x_2) = 0\}$, which is the vertical axis, separates the real plane. Thus the resulting set $\mathbb{R}^2 \setminus \{(x_1, x_2) \in \mathbb{R}^2 | x_1 = 0\}$ consists of two disjoint components.

**Example 4.12.** Recall Example 4.10. We now regard $t$ as a complex variable and so consider $\{(t, z)| H(t, z) = 0\} \subset \mathbb{C}^2$. Due to the Implicit Function Theorem we only have a path locally at a point if the determinant of the Jacobian is nonzero at this point. The points that are not regular satisfy at least one of the next two equations,

$$((1 - 2t)z^2 + 6t - 1)(1 - 2t) = 0,$$

$$2z(1 - 2t)(1 - 2t) = 0.$$ (9)

Points at which our path is interrupted are given by

$$1 - 2t = 0.$$ (10)

The only solution to (9) is $(\frac{1}{6}, 0)$ and the solution to (10) is $\{t = \frac{1}{2}\}$. The union of the solution sets to the two equations is exactly the solution set of the following system of equations,

$$((1 - 2t)z^2 + 6t - 1)(1 - 2t) = 0,$$

$$(2z(1 - 2t))(1 - 2t) = 0.$$ (11)

**Theorem 4.11** now implies that the complement of the solution set to system (11) is pathwise connected. In other words, we can find a path between any two points without running into problematic points. To walk around those problematic points, we define a new homotopy by multiplying the start polynomial $z^2 - 1$ by $e^{i\gamma}$ for a random $\gamma \in [0, 2\pi)$:

$$H(t, z) = t(5 - z^2) + e^{i\gamma}(1 - t)(z^2 - 1) = (e^{i\gamma} - t - te^{i\gamma})z^2 + te^{i\gamma} - e^{i\gamma} + 5t.$$ (12)

Now we obtain $D_zH = 2(e^{i\gamma} - t - te^{i\gamma})z$ which has $z = 0$ as its only root if $e^{i\gamma} \notin \mathbb{R}$ and $t \in [0, 1]$. Furthermore if $e^{i\gamma} \notin \mathbb{R}$ then $H(t, 0) = te^{i\gamma} - e^{i\gamma} + 5t \neq 0$ for all $t \in [0, 1]$. Additionally the coefficient of $z^2$ in (12) does not vanish for $t \in \mathbb{R}$ and thus $H(t, x) = 0$ has always two solutions for $t \in [0, 1]$ due to the Fundamental Theorem of Algebra. Therefore this so-called gamma trick yields only paths that are not interrupted and are
regular. Figure 7 displays the two paths; the left graph shows the paths in three dimensions, the right graph shows a projection of the paths on \( \mathbb{C} \). It remains to check how strict the condition \( e^{i\gamma} \notin \mathbb{R} \) is. We know \( e^{i\gamma} \in \mathbb{R} \iff \gamma = k\pi \) for \( k \in \mathbb{N} \). Since \( \gamma \in [0, 2\pi) \) these are only two points. Thus for a randomly chosen \( \gamma \) the paths exist and are regular with probability one.

The insight from this simple example applies in general. By randomly choosing the coefficient \( e^{i\gamma} \) in the starting system, the solution paths of our homotopy can (generically) circumvent all problematic points.

### 4.1.3 Multivariate Systems of Polynomial Equations

The Fundamental Theorem of Algebra does not have a multivariate analog. So, unlike the case of univariate polynomials, we do not know a priori the number of complex solutions. However, we can determine upper bounds on the number of solutions. For the sake of our discussion in this paper, it suffices to introduce the simplest such bound.

#### Definition 4.13.

Let \( F = (f_1, \ldots, f_n) : \mathbb{C}^n \to \mathbb{C}^n \) be a polynomial function. Then the number

\[
\text{deg} = \prod_i \text{deg} f_i
\]

is called the total degree or Bezout number of \( F \).

#### Theorem 4.14 (Bezout's Theorem).

Let \( d \) be the Bezout number of \( F \). Then the polynomial system \( F = 0 \) has at most \( d \) isolated solutions counting multiplicities.

This bound is tight; in fact, García and Li (1980) show that generic polynomial systems have exactly \( d \) distinct isolated solutions. But this result does not provide any guidance
for specific systems. All systems arising in applications are highly non-generic, especially since most of their coefficients are zero. We discuss how to exploit this structure below.

Next we address the difficulties we observed in Example 4.10 for the multivariate case. Consider a square polynomial system $F = (f_1, \ldots, f_n) = 0$ with $d_i = \deg f_i$. Construct a start system $G = (g_1, \ldots, g_n) = 0$ by setting

$$g_i(z) = z_i^{d_i} - 1. \quad (13)$$

Note that the polynomial $g_i(z)$ only depends on the variable $z_i$ and has the same degree as $f_i(z)$. Thus $F$ and $G$ have the same Bezout number. Now let $H = (h_1, \ldots, h_n) : \mathbb{C} \times \mathbb{C}^n \to \mathbb{C}^n$ be homotopy from the square polynomial system $F(z) = 0$ to the start system $G(z) = 0$, such that the function is linear in the homotopy parameter $t$. As a result we can write the individual elements of the homotopy as $h_i(z)$ with degree $d_i$ in the variables $z_1, \ldots, z_n$ and its coefficients being linear functions in $t$,

$$h_i(z) = \sum_{j=0}^{d_i} \left( \sum_{i_1 + \ldots + i_n = j} a_{i_1, i_2, \ldots, i_n}(t) \prod_{k=1}^{n} z_k^{i_k} \right).$$

Denote by $a_i(t)$ the product of the coefficients of the highest degree monomials of $h_i(z)$. As before, non-regular points are solutions to the following system of equations.

$$h_i = 0 \quad \forall i$$

$$\det (D_z H) \prod_i a_i(t) = 0. \quad (15)$$

Additionally, values of the homotopy parameter for which one or more of our paths might get interrupted are all $t$ that satisfy the following equation,

$$\prod_i a_i(t) = 0. \quad (15)$$

For a $t'$ satisfying the above equation it follows that the polynomial $H(t', z)$ has a lower Bezout number than $F(z)$.\footnote{Note that after homogenization, which we introduce in Section 4.2.1, this no longer poses any problem.} As in Example 4.12, we can cast (14) and (15) in a single system of equations,

$$h_i \prod_j a_j(t) = 0 \quad \forall i,$$

$$\det (D_z H) \prod_i a_i(t) = 0. \quad (16)$$

Theorem 4.11 states that the complement of the solution set to this system of equations is a pathwise connected set. So we can “walk around” those points that cause difficulties for the path-following algorithm. In fact, if we choose our paths randomly just as in Example 4.12, then we do not encounter those problematic points with probability one.
Theorem 4.15 (Gamma trick). Let $G(z) : \mathbb{C}^n \rightarrow \mathbb{C}^n$ be our start system and $F(z) : \mathbb{C}^n \rightarrow \mathbb{C}^n$ our target system. Then for almost all choices of the constant $\gamma \in [0, 2\pi)$, the homotopy

$$H(t, z) = e^{\gamma i}(1 - t)G(z) + tF(z)$$

(17)

has regular solution paths and $|\{z \mid H(t_1, z) = 0\}| = |\{z \mid H(t_2, z) = 0\}|$ for all $t_1, t_2 \in [0, 1)$. We say that a path diverges to infinity at $t = 1$ if $\|z(t)\| \rightarrow \infty$ for $z(t)$ satisfying $H(t, z(t)) = 0$ as $t \rightarrow 1$ where $\|\cdot\|$ denotes the Euclidean norm.

The Gamma trick leads to the following theorem.

Theorem 4.16. Consider the homotopy $H$ as in (17) with a start system as in (13). For almost all parameters $\gamma \in [0, 2\pi)$, the following properties hold.

1. The preimage $H^{-1}(0)$ consists of $d$ regular paths, i.e., no paths cross or bend backwards.
2. Each path either diverges to infinity or converges to a solution of $F(z) = 0$ as $t \rightarrow 1$.
3. If $\bar{z}$ is an isolated solution with multiplicity $m$, then there are $m$ paths converging to it.

Theorem 4.16 implies that the homotopy $H$ gives rise to $d$ distinct paths; each isolated root of $F$ is found by at least one path. So, we can find all isolated roots of $F$ by following all paths in $H^{-1}(0)$.

4.2 Advanced Features of All-Solution Homotopy Methods

The solution approach based on Theorem 4.16 has two significant weaknesses; the diverging paths are a major strain on the numerical method, and the Bezout number of a polynomial system, that is, the number of homotopy paths, grows exponentially in the number of variables. We can resolve the first issue by compactifying the space the homotopy is operating in. Such a compactification requires the notion of homogeneous polynomials. Section 4.2.1 describes this approach. The second issue is trickier. We can choose between two different avenues to deal with the growing number of paths. First, there exist different (often tighter) bounds on the number of solutions than the Bezout number. These tighter bounds allow us to cut down on the number of diverging paths. We briefly address this idea in Section 4.2.2. Secondly, we can take advantage of the known structure of the polynomial $F$ to reduce the number of paths, that is, we can use prior knowledge on similar systems to reduce the number of paths we must follow. We explain this approach in some detail in Section 4.2.3.

4.2.1 Homogenization and Projective Space

The all-solution homotopy method presented in Section 4.1.3 has the unattractive feature that it must follow diverging paths. This requirement leads to numerical difficulties. Homogenization of the polynomials reformulates solutions “at infinity” as a possible finite solution. Thus paths that have been diverging will now converge to these solutions. This transformation does not eliminate these paths but it stabilizes the numerical methods.
Definition 4.17. The homogenization $\hat{f}_i(z_0, z_1, \ldots, z_n)$ of the polynomial $f_i(z_1, \ldots, z_n)$ of degree $d_i$ is defined by

$$\hat{f}_i(z_0, z_1, \ldots, z_n) = z_0^{d_i} f_i\left(\frac{z_1}{z_0}, \ldots, \frac{z_n}{z_0}\right).$$

Effectively, each term of $\hat{f}_i$ is obtained from multiplying the corresponding term of $f_i$ by the power of $z_0$ that leads to a new degree of that term of $d_i$. So, if the term originally had degree $d_{ij}$ then it is multiplied by $\frac{d_i}{d_i-d_{ij}}$. Performing this homogenization for each polynomial $f_i$ in the system

$$F(z_1, \ldots, z_n) = 0 \quad (18)$$

leads to the transformed system

$$\hat{F}(z_0, z_1, \ldots, z_n) = 0. \quad (19)$$

We illustrate homogenization by an example.

Example 4.18. Recall the polynomials from Example 2.17, with the three unknowns denoted by $z_1, z_2, z_3$, respectively,

$$z_1 - z_2 z_3^3 - 2z_3^3 + 1 = 0,$$
$$-z_1 + z_2 z_3 - 3z_3 + 4 = 0,$$
$$z_1 + z_2 z_3^9 = 0.$$

The three polynomials are of degrees 4, 2, and 10, respectively. We multiply the first polynomial by $z_0^4$ and replace each $z_i$ by $z_i/z_0$ and obtain

$$z_0^4 \left(\frac{z_1}{z_0} - \frac{z_2}{z_0} \left(\frac{z_3}{z_0}\right)^3\right) - 2 \left(\frac{z_3}{z_0}\right)^3 + 1 = z_0^3 z_1 - z_2 z_3^3 - 2z_0 z_3^3 + z_0^4.$$

Observe that each individual monomial of the homogenized polynomial is of identical degree 4. The complete homogenized system then appears as follows,

$$z_0^3 z_1 - z_2 z_3^3 - 2z_0 z_3^3 + z_0^4 = 0,$$
$$-z_0 z_1 + z_2 z_3 - 3z_0 z_3 + 4z_0^2 = 0,$$
$$z_0^9 z_1 + z_2 z_3^9 = 0.$$

For convenience we use the notation $\hat{z} = (z_0, z_1, \ldots, z_n)$ and write $\hat{F}(\hat{z}) = 0$. By construction, all polynomials $\hat{f}_i, i = 1, \ldots, n$, are homogeneous and so for any solution $\hat{b}$ of $\hat{F}(\hat{z}) = 0$ it holds that $\hat{F}(\lambda \hat{b}) = 0$ for any complex scalar $\lambda \in \mathbb{C}$. So, the solutions to system (19) are complex lines through the origin in $\mathbb{C}^{n+1}$.
Theorem 4.21 (Bezout’s theorem in projective space \( \mathbb{P}^n \)). If system (19) has only a finite number of solutions in \( \mathbb{P}^n \) and if \( d \) is the Bezout number of \( F \), then it has exactly \( d \) solutions (counting multiplicities) in \( \mathbb{P}^n \).

If we view the system of Eq. (19) in affine space \( \mathbb{C}^{n+1} \) instead of in complex projective space \( \mathbb{P}^n \) then it is actually underdetermined because it consists of \( n \) equations in \( n + 1 \) unknowns. For a computer implementation of a homotopy method, however, we need a determinate system of equations. For this purpose we add a simple normalization. Using the described relationship between solutions of the two systems (18) and (19) we can now introduce a third system to find the solutions of system (18). Define a new linear
function
\[ u(z_0, z_1, \ldots, z_n) = \xi_0 z_0 + \xi_1 z_1 + \cdots + \xi_n z_n \]
with random coefficients \( \xi_i \in \mathbb{C} \). The normalization line is parallel to a solution “line” in non-generic cases; a random choice of coefficients \( \xi_i \in \mathbb{C} \) prohibits this case with probability one. Now define
\[
\tilde{f}_i(z_0, z_1, \ldots, z_n) := \hat{f}_i(z_0, z_1, \ldots, z_n), \quad i = 1, \ldots, n,
\]
\[
\tilde{f}_0(z_0, z_1, \ldots, z_n) := u(z_0, z_1, \ldots, z_n) - 1.
\] (20)

The resulting system of equations
\[
\tilde{F} = (\tilde{f}_0, \tilde{f}_1, \ldots, \tilde{f}_n) = 0
\] (21)
has \( n + 1 \) equations in \( n + 1 \) variables. Note that the system \( \tilde{F}(\hat{z}) \) has the same total degree \( d \) as the system \( F(\hat{z}) \) in the original system of Eq. (18). As a start system we choose
\[
G_i(z_0, z_1, \ldots, z_n) = z_i^d - z_0^d, \quad i = 1, \ldots, n,
\]
\[
G_0(z_0, z_1, \ldots, z_n) = u(z_0, z_1, \ldots, z_n) - 1.
\] (22)

We write the resulting system as \( G(\hat{z}) = 0 \) and define the homotopy
\[
H(t, \hat{z}) = t\tilde{F}(\hat{z}) + e^{\gamma i}(1 - t)G(\hat{z})
\] (23)
for a \( \gamma \in [0, 2\pi) \). To illustrate a possible difficulty with this approach we examine the system of Eqs. 1–3 that we derived for the Bertrand pricing game in Section 3.1.1.

**Example 4.22.** After homogenization of the equilibrium system (1, 2, 3) in the variables \( p_x, p_y, \) and \( Z \) with the variable \( x_0 \) we obtain the following polynomial equations.

\[
0 = -p_x^2 x_0^4 - p_y^2 y_0^4 + Z^2 p_x^2 p_y^2,
\]
\[
0 = -2700x_0^{10} + 2700p_x x_0^9 + 8100Z^2 p_x^2 x_0^6 - 5400Z^2 p_x^3 x_0^5 + 51Z^3 p_x^4 x_0^4 - 2Z^3 p_x^7,
\]
\[
0 = -2700x_0^{10} + 2700p_y x_0^9 + 8100Z^2 p_y^2 x_0^6 - 5400Z^2 p_y^3 x_0^5 + 51Z^3 p_y^4 x_0^4 - 2Z^3 p_y^7.
\]
The solutions at infinity are those for which \( x_0 = 0 \). In this case the system simplifies as follows
\[
Z^2 p_x^2 p_y^2 = 0, \quad -2Z^3 p_x^7 = 0, \quad -2Z^3 p_y^7 = 0.
\]
After setting \( Z = 0 \) all equations hold for any values of \( p_x \) and \( p_y \). There is a continuum of solutions at infinity. Such continua can cause numerical difficulties for the path-following procedure.

The previous example shows that we do not consider the true compactification of our variety. And, in general, we cannot obtain it by simply homogenizing the generators of the ideal. However, if the ideal is given by a Gröbner basis, then the compactification
can be obtained by looking at the homogenized versions of the generators. But clearly this often is too costly to do.

The following theorem now states that in spite of the previous example our paths converge to the relevant isolated solutions.

**Theorem 4.23.** Let the homotopy $H$ be as in (23) with Bezout number $d$. Then the following statements hold for almost all $\gamma \in [0, 2\pi)$:

1. The homotopy has $d$ continuous solution paths.
2. Each path either converges to an isolated non-singular or to a singular\textsuperscript{11} solution, i.e., one where the rank of the Jacobian drops.
3. If $b$ is an isolated solution with multiplicity $m$, then there are $m$ paths converging to it.
4. Along the paths the homotopy parameter $t$ is monotonically increasing, i.e., the paths do not bend backward.

Now we can apply the homotopy $H$ as defined in Eq. (23) and find all solutions of the system (21). There will be no diverging paths. From the solutions of (21) we obtain the solutions of the original system (18).

An additional advantage of the above approach lies in the possibility to scale our solutions via $u$. If a solution component $z_i$ becomes too large, then this will cause numerical problems, e.g., the evaluation of polynomials at such a point becomes rather difficult. Thus, if something like this happens we pick a new set of $\xi_i$. Furthermore, we eliminated the special case of infinite paths and we do not have to check whether the length of the path grows too large. Instead every diverging path has become a converging one. So while tracking a path we do not need to check whether the length of the path exceeds a certain bound.

Theoretically we have eliminated the problem of solutions at infinity. However, to decide whether a path diverges, we still have to decide if $b_0$ is in fact equal to 0. Since we only determine the solution up to the numerical precision, this still leaves a potential for prematurely truncating the path.

**4.2.2 The $m$-Homogeneous Bezout Number**

The number of paths $d$ grows rapidly with the degree of individual equations. For many economic models, we may expect that there are only a few equilibria, that is, our systems have few real solutions and usually even fewer economically meaningful solutions. As a result we may have to follow a large number of paths that do not yield useful solutions. As we have seen in Example 4.22, there may be continua of solutions at infinity which can cause numerical difficulties. Therefore it would be very helpful to reduce the number of paths that must be followed as much as possible.

\textsuperscript{11} This might be an isolated root with multiplicity higher than one, e.g., a double root of the system $F$, or a non-isolated solution component as in Examples 4.22 and 3.4.
We will present two approaches for a reduction in the number of paths. The first approach sets the homogenized polynomial system not into \( \mathbb{P}^n \) but in a product of \( m \) projective spaces \( \mathbb{P}^n_1 \times \cdots \times \mathbb{P}^n_m \). For this purpose the set of variables is split into \( m \) groups. In the homogenization of the original polynomial \( F \) each group of variables receives a separate additional variable; thus this process is called \( m \)-homogenization. The resulting bound on the number of solutions, called the \( m \)-homogeneous Bezout number, is often much smaller than the original bound and thus leads to the elimination of paths tending to solutions at infinity. In this paper we do not provide details on this approach but only show its impact in our computational examples. We refer the interested reader to Sommese and Wampler (2005) and the citations therein. The first paper to introduce \( m \)-homogeneity appears to be Morgan and Sommese (1987).

The second approach to reduce the number of paths is the use of parameter continuation homotopies. This approach is well suited for economic applications.

### 4.2.3 Parameter Continuation Homotopy

Economic models typically make use of exogenous parameters such as endowments, price elasticities, cost coefficients, or many other pre-specified constants. Often we do not know the exact values of those parameters and so would like to solve the model for a variety of different parameter values. Clearly solving the model each time “from scratch” will prove impractical whenever the number of solution paths is very large. The parameter continuation homotopy approach enables us to greatly accelerate the repeated solution of an economic model for different parameter values. After solving one instance of the economic model we can construct a homotopy that alters the parameters from their previous to their new values and allows us to track solutions paths from the previous solutions to new solutions. Therefore, the number of paths we need to follow is greatly reduced.

The parameter continuation approach rests on the following theorem which is a special case of a more general result; see Sommese and Wampler (2005, Theorem 7.1.1).

**Theorem 4.24 (Parameter Continuation).** Let \( F(z, q) = (f_1(z, q), \ldots, f_n(z, q)) \) be a system of polynomials in the variables \( z \in \mathbb{C}^n \) with parameters \( q \in \mathbb{C}^m \),

\[
F(z, q) : \mathbb{C}^n \times \mathbb{C}^m \rightarrow \mathbb{C}^n.
\]

Additionally let \( q_0 \in \mathbb{C}^m \) be a point in the parameter space, where \( k = \max_q \{|z \mid F(z, q) = 0; \det \left( \frac{\partial F}{\partial z} (z, q_0) \right) \neq 0 \}| \) is the number of non-singular isolated solutions. For any other choice of parameters \( q_1 \) and a random \( \gamma \in [0, 2\pi) \) define

\[
\varphi(s) = e^{i\gamma} s (s-1) + sq_1 + (1-s)q_0.
\]

Then the following statements hold.

1. \( k = |\{z \mid F(z, q) = 0; \det \left( \frac{\partial F}{\partial z} (z, q) \right) \neq 0 \}| \) for almost all \( q \in \mathbb{C}^n \).
2. The homotopy \( F(z, \varphi(s)) = 0 \) has \( k \) non-singular solution paths for almost all \( \gamma \in [0, 2\pi) \).
3. For almost all $\gamma \in [0, 2\pi)$, all-solution paths converge; in addition, each isolated non-singular solution of $F(z, \varphi(1)) = 0$ has a path converging to it.

The theorem has an immediate practical implication. Suppose we already solved the system $F(z, q_0) = 0$ for some parameter vector $q_0$. Under the assumption that this system has the maximal number $k$ of locally isolated solutions across all parameter values, we can use this system as a start system for solving the system $F(z, q_1) = 0$ for another parameter vector $q_1$. The number of paths that need to be tracked is $k$ instead of the Bezout number $d$ or some $m$-homogeneous Bezout number. In many applications $k$ is much smaller (sometimes orders of magnitude smaller) than these upper bounds. As a result the parameter continuation homotopy dramatically reduces the number of paths that we must track. More importantly, no path ends at a solution at infinity for almost all $q_1 \in \mathbb{C}^n$. As we observe in our examples, exactly these solutions often create numerical problems for the path-tracking software, in particular if there are continua of solutions at infinity as in Example 4.22. And due to those numerical difficulties the running times for tracking these paths are often significantly larger than for tracking paths that end at finite solutions. In sum, the parameter continuation homotopy approach has the potential to be of great importance for finding all equilibria of economic models.

A statement similar to that of Theorem 4.24 holds if we regard isolated solutions of some fixed multiplicity. But we then have to track paths which have the same multiplicity. Tracking such paths requires a lot more computational effort than non-singular paths. The homotopy continuation software BERTINI enables the user to track such paths since it allows for user-defined parameter continuation homotopies.

## 5. APPLYING HOMOTOPY METHODS

We briefly describe the software package BERTINI and the potential computational gains from a parallel version of the software code.

### 5.1 Software

#### 5.1.1 BERTINI

The software package BERTINI, written in the programming language C, offers solvers for a few different types of problems in numerical algebraic geometry; see Bates et al. (2005). The most important feature for our purpose is BERTINI’s homotopy continuation routine for finding all isolated solutions of a square system of polynomial equations. In addition to an implementation of the advanced homotopy of Theorem 4.23 (see Section 4.2.1) it also allows for $m$-homogeneous start systems as well as parameter continuation homotopies as in Theorem 4.24; see Sections 4.2.2 and 4.2.3. BERTINI has an intuitive interface which allows the user to quickly implement systems of polynomial equations; see Sections 5.2.1 and 5.2.2 for the type of code that a user must supply. BERTINI can be downloaded free of charge under [http://www3.nd.edu/~sommese/bertini/](http://www3.nd.edu/~sommese/bertini/).
5.1.2 Other Software Packages
Two other all-solution homotopy software packages are PHCpack (Verschelde, 1999, 2011) written in ADA and POLSYS_PLP (Wise et al., 2000) written in FORTRAN90 and which is intended to be used in conjunction with HOMPACK90 (Watson et al., 1997), a popular homotopy path solver. Because of its versatility, stable implementation, great potential for parallelization on large computer clusters, and friendly user interface, we use Bertini for all our calculations.

5.1.3 Parallelization
The overall complexity of the all-solution homotopy method is the same as for other methods used for polynomial system solving. A major advantage of this method, however, is that it is naturally parallelizable. Following each path is a distinct task, i.e., the paths can be tracked independently from each other. Moreover, the information gathered during the tracking process of a path cannot be used to help track other paths.

The software package Bertini is available in a parallel version. As of this writing, we have already successfully computed examples via parallelization on 200 processors at the CSCS cluster (Swiss Scientific Computing Center).

5.2 Bertrand Pricing Game Continued
We return to the duopoly price game from Section 3.1 and show how to solve the problem with Bertini. We also show how to apply some of the advanced features from Section 4.2.

5.2.1 Solving the Bertrand Pricing Game with Bertini
To solve the system 1–3 from Section 3.1 in Bertini, we write the following input file:

```plaintext
CONFIG
MPTYPE: 2;
END;

INPUT
variable_group px, py, z;
function f1, f2, f3;
f1 = -(px^2) - py^2 + z^2 * px^2 * py^2;
f2 = -(2700) + 2700 * px + 8100 * z^2 * px^2 - 5400 * z^2 * px^3 + 51 * z^3 * px^6 - 2 * z^3 * px^7;
f3 = -(2700) + 2700 * py + 8100 * z^2 * py^2 - 5400 * z^2 * py^3 + 51 * z^3 * py^6 - 2 * z^3 * py^7;
END;
```

The option MPTYPE:2 indicates that we are using adaptive precision path-tracking. The polynomials f1, f2, f3 define the system of equations. The Bezout number is \(6 \times 10 \times 10 = 600\). Thus, Bertini must track 600 paths. We obtain 18 real and 44 complex
solutions and we also have 538 truncated infinite paths. BERTINI lists the real solution in the file real_finite_solutions and all finite ones in finite_solutions.

Next we show how to reduce the number of paths with $m$-homogenization (see Section 4.2.2). Replace the command

```
variable_group px,py,z;
```

by

```
variable_group px;
variable_group py;
variable_group z;
```

By separating the variables in the different groups, we indicate how to group them for the $m$-homogenization. As a result we have only 182 paths to track. Doing so, we find the same 18 real and 44 complex solutions as before, but now only 120 paths converge to solutions at infinity.

### 5.2.2 Application of Parameter Continuation

To demonstrate parameter continuation, we choose $n$ as the parameter and vary it from 2700 to 1000. Note that in BERTINI the homotopy parameter goes from 1 to 0. So, we define a homotopy just between those two values,

$$ n = 2700t + (0.22334546453233 + 0.974739352i)t(1 - t) + 1000(1 - t). $$

Thus for $t = 1$ we have $n = 2700$ and if $t = 0$ then $n = 1000$. The complex number in the equation is the application of the Gamma trick. We also have to provide the solutions for our start system. We already solved this system. We just rename BERTINI’s output file finite_solutions to start which now provides BERTINI with the starting points for the homotopy paths. In addition, we must alter the input file as follows.

```
CONFIG
    USERHOMOTOPY: 1;
    MPTYPE: 2;
END;

INPUT
    variable px,py,z;
    function f1, f2, f3;
    pathvariable t;
    parameter n;
    n = t*2700
    +(0.22334546453233 + 0.974739352*1.0)*t*(1-t)+(1-t)*1000;
    f1 = -(px^2)-py^2+z^2*px^2*py^2;
    f2 = -(n)+n*px+3*n*z^2*px^2-2*n*z^2*px^3
         +51*z^3*px^6-2*z^3*px^7;
    f3 = -(n)+n*py+3*n*z^2*py^2-2*n*z^2*py^3
         +51*z^3*py^6-2*z^3*py^7;
END;
```

If we run BERTINI we obtain 14 real and 48 complex solutions. Note that the number of real solutions has dropped by 4. Thus if we had not used the Gamma trick some of our
paths would have failed. There are only five positive real solutions. The first three solutions in Table 2 fail the second-order conditions for at least one firm. The fourth solution fails the global-optimality test. Only the last solution in Table 2 is an equilibrium for the Bertrand game for $n = 1000$.

### 5.2.3 The Manifold of Real Positive Solutions

The parameter continuation approach allows us to compare solutions and thus equilibria for two different (vectors of) parameter values $q_0$ and $q_1$ of our economic model. Ideally we would like to push our analysis even further and, in fact, compute the equilibrium manifold for all convex combinations $sq_1 + (1 - s)q_0$ with $s \in [0, 1]$. Observe that Theorem 4.24 in Section 4.2.3 requires a path between $q_0$ and $q_1$ of the form

$$\varphi(s) = e^{i\gamma} s (s - 1) + sq_1 + (1 - s)q_0$$

with a random $\gamma \in [0, 2\pi)$. Note that for real values $q_0$ and $q_1$ the path $\varphi(s)$ is not real and so all solutions to $F(z, \varphi(s)) = 0$ are economically meaningless for $s \in (0, 1)$. This problem would not occur if we could drop the first term of $\varphi(s)$ and instead use the convex combination

$$\tilde{\varphi}(s) = sq_1 + (1 - s)q_0$$

in the definition of the parameter continuation homotopy. Now an examination of the real solutions to $F(z, \tilde{\varphi}(s)) = 0$ would provide us with the equilibrium manifold for all $\tilde{\varphi}(s)$ with $s \in [0, 1]$. Unfortunately, such an approach does not always work. While the number of isolated finite solutions remains constant with probability one, the number of real solutions may change. A parameter continuation homotopy with $\tilde{\varphi}(s)$ does not allow for this change. Judd et al. (2012) explore the involved issues in more detail.

In sum, we observe that a complete characterization of the equilibrium manifold is not a simple exercise. When we employ the parameter continuation approach with a path of parameters in real space, then we have to allow for the possibility of path-tracking failures whenever the number of real and complex solution changes. The determination of the entire manifold of positive real solutions may, therefore, require numerous homotopy runs. Despite these difficulties we believe that the parameter continuation approach is a very helpful tool for the examination of equilibrium manifolds.

### 5.3 Walrasian Exchange Economy

We reexamine Example 3.4. The Bertini input appears as follows.

```plaintext
CONFIG
MPTYPE: 2;
```
FINALTOL: 1e-18;
END;

INPUT
variable_group x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12;
variable_group x13,x14,x15;
function f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13,f14,f15;
f1=-30*(0+x2)*x13+(1+x1);
f2=-30*(0+x3)*x14+(1+x1);
f3=-30*(0+x4)*x15+(1+x1);
f4=-(1+x6)*x13+30*(0+x5);
f5=-(0+x7)*x14+(0+x5);
f6=-(0+x8)*x15+(0+x5);
f7=-(0+x10)*x13+(0+x9);
f8=-(1+x11)*x14+30*(0+x9);
f9=-(0+x12)*x15+(0+x9);
f10=-(0-x2-x6-x10)*x13+(0-x1-x5-x9);
f11=-(0-x3-x7-x11)*x14+(0-x1-x5-x9);
f12=-(1-x4-x8-x12)*x15+30*(0-x1-x5-x9);
f13=x1+x2*x13^3+x3*x14^3+x4*x15^3;
f14=x5+x6*x13^3+x7*x14^3+x8*x15^3;
f15=x9+x10*x13^3+x11*x14^3+x12*x15^3;
END;

Solving this system of equations, we obtain 15 real and 20 complex solutions. The complex solutions are also singular. Recall from Example 3.4 that all complex solutions lie in a one-dimensional set. Unlike with the Gröbner basis, we do not have to remove the complex component. Instead we know that the homotopy algorithm, in theory, will find all isolated solutions. However, we no longer receive a certificate that the one-dimensional solution component contains no real solutions.

5.4 Homotopy Continuation Compared to Gröbner Basis

In this section we want to give a short comparison between the potentially exact Gröbner bases methods and the purely numerical homotopy methods. Both algorithmic approaches have a double exponential run time. So from the theoretic complexity standpoint it does not matter which approach we use. However, in practice, there is a performance difference.

The homotopy approach has a performance advantage over the symbolic approach. One of the reasons is the use of floating point arithmetic. The main focus of computer scientists and software developers has been to optimize computers for floating point operations. As a consequence the system can also be given by approximate data. Furthermore
the path-tracking problem is easily parallelizable, which is not possible for Buchberger’s algorithm. These facts provide homotopy methods with a clear performance edge over the exact arithmetic of symbolic Gröbner bases methods. To illustrate this point, we consider the following example.

```
CONFIG
MPTYPE: 2;
FINALTOL: 1e-18;
END;

INPUT
variable_group x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12;
variable_group x13,x14,x15;
function f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12,f13,f14,f15;

f1= -30*(1+x2*x1)*x13+(1+x1*x2);
f2=-30*(2+x3*x5)*x14+(1+x1*x5);
f3=-30*(3+x4)*x15+(1+x1)+1;
f4=-(1+x6)*x13+30*(0+x5)+5;
f5=-(0+x7)*x14+(0+x5)+1;
f6=-(0+x8)*x15+(0+x5)+2;
f7=-(0+x10)*x13+(0+x9)+3;
f8=-(1+x11)*x14+30*(0+x9)+x1^2+x2+x3*x4^2;
f9=-(0+x12)*x15+(0+x9)-1;
f10=-(2-x2-x6-x10)*x13+(0-x1-x5-x9)-2;
f11=-(1-x3-x7-x11)*x14+(0-x1-x5-x9)-3;
f12=-(1-x4-x8-x12)*x15+30*(0-x1-x5-x9);
f13= x1+x2*x13^3+x3*x14^3+x4*x15^3+1;
f14= x5+x6*x13^3+x7*x14^3+x8*x15^3+2;
f15= x9+x10*x13^3+x11*x14^3+x12*x15^3+3;
END;
```

Current computer technology allows us to solve this problem with `{ber}. However, the use of floating point leads to the potential problem that we may not have found all solutions. The path tracker may have jumped from some path to a solution to another path which leads to infinity. While `{ber} allows the use to adjust the precision of the path-following, this feature does not provide a theoretical warranty that all solutions have been found. On the contrary, the symbolic Gröbner bases methods do provide us with an exact count of the number of solutions.
6. CONCLUSION

Multiplicity of equilibria is a common problem in many economic models. Often equilibria of economic models are characterized as solutions to a system of polynomial equations. Therefore, methods that allow the computation of all solutions to such systems are of great interest to economists. In this chapter, we have provided an overview on the application of Gröbner bases methods and all-solution homotopy methods to finding all solutions of polynomial systems of equations. Several examples have shown how easy it is to use modern software packages to apply these methods to economic problems. It is our hope that this chapter will motivate economists to pay much more attention to equilibrium multiplicity than they have given this important issue in the past.

ACKNOWLEDGMENTS

We thank two referees for detailed comments on an earlier version. Over the years we received many useful questions from students at ICE summer workshops at the University of Chicago and ZICE workshops at the University of Zurich which influenced the presentation of the material in this chapter. Felix Kubler and Karl Schmedders gratefully acknowledge financial support from the Swiss Finance Institute. Felix Kubler also acknowledges support from the ERC.

REFERENCES


