Some Computational Methods for Systems of Nonlinear Equations and Systems of Polynomial Equations

W. FORSTER
Department of Mathematics, University of Southampton, Southampton, England.

(Received: 24 September 1990; accepted: 27 November 1991)

Abstract. This paper gives a brief survey and assessment of computational methods for finding solutions to systems of nonlinear equations and systems of polynomial equations. Starting from methods which converge locally and which find one solution, we progress to methods which are globally convergent and find an *a priori* determinable number of solutions. We will concentrate on simplicial algorithms and homotopy methods. Enhancements of published methods are included and further developments are discussed.

Key words. Systems of nonlinear equations, systems of polynomial equations, simplicial algorithms, pivoting algorithms, homotopy algorithms, Kuhn's method for finding roots of a polynomial, Nielsen fixed point theory.

1. Introduction

Systems of nonlinear equations play an important role in modelling scientific problems, engineering problems and commercial problems. These applications provide a large source of systems of nonlinear equations. If we wish, e.g., to solve a linear or nonlinear optimization problem, applying the Kuhn–Tucker conditions, we end up with a system of nonlinear equations. The motion of robot arms, rendering of surfaces in CAD, chemical reactions, and many other applications are sources of nonlinear equations.

Due to the difficulties encountered in trying to solve systems of nonlinear equations, such systems have always been one of the frontiers of research and have attracted the best minds. Let us look at some old and new statements about the solution of nonlinear equations.

In 1958 Hartree ([43], p. 233) writes about nonlinear algebraic equations in three or more variables: "There is no satisfactory practical method, graphical or tabular, of displaying the behaviour of functions of three or more variables, and the appropriate location of solutions of such equations is therefore difficult." This was written before libraries of computational algorithms like NAG or IMSL were created. But has anything changed?

In 1983 we read in the IMSL Reference Edition ([78], p. 240): "Solving systems of nonlinear equations is perhaps the most difficult problem in all of numerical

Journal of Global Optimization **2**: 317–356, 1992.

computation. It is not unheard of for even 5 equations to be very difficult; one can 'hide' a solution in 5-dimensional space rather easily."

Has further insight and recent activity and experience with computational methods contributed to more optimism? It seems not. Let us quote from a widely used collection of algorithms published in 1989 ([74], p. 305): "We make an extreme, but wholly defensible statement: There are no good, general methods for solving systems of more than one nonlinear equation. Furthermore, it is not hard to see there will never be any good, general methods."

In this paper we will give a brief survey of computational methods for systems of nonlinear equations. We will show that the situation is not as bleak as above quotations make us believe. We will show how the additional problem of having more than one solution to systems of nonlinear equations has mobilized the imagination of researchers. We will show how entirely new fields of mathematics, like algebraic geometry, algebraic topology and others (not usually associated with computational methods), have been employed as new resources. We will show how these new techniques have led to novel methods, which not only exhibit global convergence, but are also capable of finding more than one solution. On the other hand, we will discuss recent advances in research in a field now known as 'fractology', which not only have been given us pretty pictures, but also have made it clear that methods that have been used for centuries for solving single equations (e.g., Newton iteration) are not suitable for finding more than one solution in a controlled manner.

2. Classical Methods and Their Limitations

The most widely applied systematic methods for solving systems of nonlinear equations are contraction mappings. The first abstract formulation of the contraction mapping principle was given by Banach [6].

A mapping $T: X \to X$ is said to be a contraction mapping if

$$\mathrm{d}(T(x), T(y)) \leqslant K\, \mathrm{d}(x, y)$$

is satisfied for all $x, y \in X$, where $0 \leqslant K < 1$, X is a metric space, and $\mathrm{d}(.\,,.)$ is a metric on X.

Krasnoselskii and his co-workers have perhaps pushed the utilization of the contraction mapping principle as far as one can go. See, e.g., [54] for fixed point theorems in a compressed cone and in an expanding cone.

The excellent book on iterative methods by Ortega and Rheinboldt [71] is utilizing mainly multivariate calculus and first derivatives. In this book we find the following remark: "The influence of the variation of the equation, the dimension, and the initial data upon the outcome of the computation is still very little understood both from a practical as well as a theoretical viewpoint . . . " We will address some of these questions later.

An excellent survey covering algorithms for finding zeros and extrema is Brent

[8]. The main mathematical tools used in this 300 page survey can be characterized by metric spaces, search methods, some differentiability, random starting points, rates of convergence, etc. Brent's survey includes computer programs.

The widely used Newton method and variations of Newton's method can be considered to be contraction mappings in the regions in which they converge. The 1970s saw extensive research into various aspects of Newton methods, especially into improving convergence.

In 1976 Smale [86] introduced "global Newton methods" in connection with price adjustment processes. His main mathematical tools are taken from differential topology and include diffeomorphisms, sets of measure zero, Sard's theorem, etc.

In 1981 Smale [87] applied global Newton methods to finding zeros of polynomials. He gave an algorithm for finding zeros of polynomials, which had a probability μ of failure (with $0 < \mu < 1$), or would find a root with probability of $1 - \mu$. The mathematical theory of polynomials of degree n guarantees the existence of n roots. An algorithm which does not mirror this mathematical result is unsatisfactory. It is inconceivable that an algorithm which finds zeros with probability $1 - \mu$, but cannot guarantee to find a zero, is ever to be used in a safety critical application (e.g., in control theory).

By 1985 the field had developed further. Newton iterations in the complex plane became famous. Mandelbrot had produced his sets and the French mathematician Gaston Julia had been rediscovered. Computer generated graphics of polynomials was published [72]. The regions of attraction, which are so useful for Newton iteration and global Newton methods, had boundaries of amazing complexity (Julia sets). These beautiful pictures with their intricate baroque ornaments as boundary of the regions of attraction must have shown every enthusiast of iterative methods, especially Newton methods, that this type of behaviour makes such algorithms unacceptable for any type of safety critical engineering application. Smale in his 1985 paper ([88], p. 97) drew the inevitable conclusion and stated: "For polynomials of higher degree, Newton's method is not generally convergent in any reasonable sense."

One of the features of nonlinear equations which distinguishes them from linear equations is the fact that in general they have more than one solution. This simple fact immediately restricts the usefulness of contraction mapping methods. Contraction mappings can only have one fixed point and in the context of nonlinear equations this means that they can only be used locally. Furthermore, not all solutions can be transformed into attractive fixed points and therefore contraction mappings cannot find them, even if we know a starting point for the iteration near the solution.

The heavy emphasis in the literature on contraction mappings cannot conceal the fact that such methods have a built-in inefficiency if one is interested in finding more than one solution, e.g., all roots of a polynomial. Even if different starting values are used, the same solution may be found several times.

Another fact so clearly pointed out in "Numerical Recipes in PASCAL" [74] is the irritating property of systems of nonlinear equations (2 or more variables) to have a solution set, which in general does not consist of isolated points but of lower dimensional manifolds (e.g., lines of solutions, hyperplanes, etc.).

Let us recall some material on ideals from algebra. If we consider, e.g., the ring of bounded continuous functions $C[a, b]$ on a closed interval $a \leq x \leq b$ and X is a subset of $[a, b]$, then the set

$$I(x) = \{ f \mid f(x) = 0 \text{ for every } x \in X \}$$

is an ideal in this ring.

If we restrict ourselves for the moment to systems of polynomials in more than one variable, then systems with isolated solution points are called systems with zero-dimensional polynomial ideals. Numerical algorithms for finding roots of such systems have been published [24]. Algebraic manipulation is capable of factoring such systems too. The difficulties start when we consider general systems, where the solution set no longer consists of isolated points. Algebraic manipulation, i.e., factoring such polynomials is considered an open problem [21]. Nevertheless, we will show how with the help of true homotopy algorithms the numerical solution of such systems can be achieved.

3. Simplicial Algorithms

The backbone of all simplicial algorithms is Sperner's lemma [90]. One version of Sperner's lemma [31, 91] can be stated in the following form.

LEMMA. *Given a subdivided n-dimensional simplex with integers* $\{0, 1, \ldots, n\}$ *attached to its vertices, then the number of* $(n-1)$*-dimensional simplices on the boundary with labels* $\{0, 1, \ldots, n-1\}$ *is equal to the number of simplices in the interior with labels* $\{0, 1, \ldots, n\}$. *The simplices are counted with orientation.*

An illustration of the lemma is given in Figure 1.

Sperner's lemma was published in 1928 and it was always clear that there is a close connection with the famous Brouwer fixed point theorem [10]. In 1972 Yoseloff [99] showed that Sperner's lemma and Brouwer's fixed point theorem are equivalent. Following Sperner's lemma, a large number of labelling lemmas has been published, e.g., Tucker [95], Ky Fan [29], van der Laan and Talman [61], Freund [36], and van der Laan *et al.* [62]. For a detailed list see Forster [31].

Given a completely labelled simplex on the boundary, it is Sperner's lemma that allows us to construct globally convergent algorithms without imposing restrictive conditions on the function (e.g., we do not require the mapping T to be a contraction mapping). From the completely labelled $(n-1)$-dimensional simplex on the boundary we construct a chain of n-dimensional simplices leading from the boundary to the completely labelled n-dimensional simplex in the interior. Structures of this kind are called pseudomanifolds.

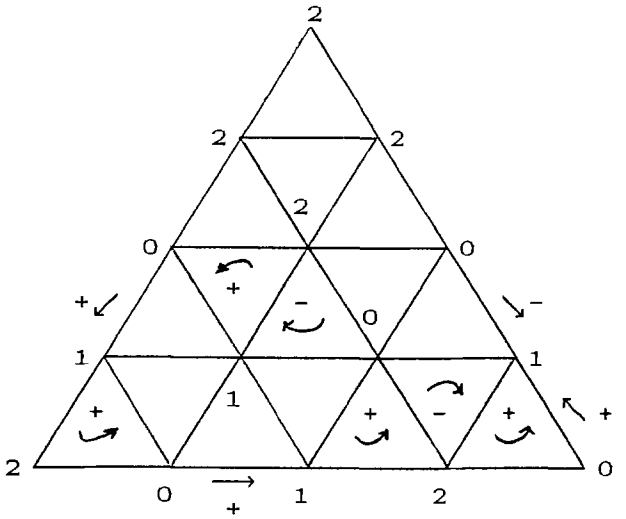

Fig. 1. Illustration of generalized Sperner lemma. Count on boundary of oriented one-dimensional simplices labelled 01 equals 2. Count in interior of oriented two-dimensional simplices labelled 012 equals 2.

The theory of simplices and other piecewise linear structures can be found in almost any book on algebraic topology, e.g. [89]. For an extensive coverage of the topology of simplices, but without any material on algorithms, see, e.g. [100, 80].

Because of its central importance for simplicial algorithms we give the definition of a pseudomanifold:

An n-dimensional pseudomanifold is a simplicial complex K such that:

(a) Every simplex of K is a face of some n-dimensional simplex of K.

(b) Every $(n-1)$-dimensional simplex of K is the face of at most two (and at least one) n-dimensional simplex of K.

(c) Any two n-dimensional simplices can be used as the first and last members of a finite sequence of n-dimensional simplices such that the intersection of any two consecutive n-dimensional simplices in the sequence have an $(n-1)$-dimensional simplex in common.

Pseudomanifolds can be constructed in many different ways. Instead of attaching integers to each vertex of a simplex, we can attach vectors (vector labelling) and obtain a pseudomanifold. Another way of constructing pseudomanifolds is due to Scarf [84] via 'primitive sets'. For an explanation of the different concepts and the connection between all these different methods of creating pseudomanifolds see, e.g., Gould and Tolle [41]. If the labelling is chosen in an appropriate manner than the completely labelled simplex corresponds to an approximate fixed point x^* of the mapping $T(x)$, i.e., $T(x^*) = x^*$.

We give an example of an integer labelling function and a vector labelling function.

EXAMPLE. Definition of standard integer labelling on an n-dimensional euclidean space $\mathbb{R}^n$:
For any $x \in \mathbb{R}^n$ let

$$i = \min_j \{ j \mid f_j(x) - x_j = \max_k (f_k - x_k) \} ,$$

$i, j, k = 1, \ldots, n$.

Then x receives the label $l(x)$ where

$$l(x) = i \quad \text{if } f_i - x_i \geqslant 0$$

and

$$l(x) = 0 \quad \text{if } f_i - x_i < 0 .$$

A completely labelled simplex (i.e., a simplex with all labels from 0 to n) then approximates a fixed point x^* of the mapping $f(x)$.

Definition of standard vector labelling on an n-dimensional euclidean space $\mathbb{R}^n$:

A point $x \in \mathbb{R}^n$ receives the $(n+1)$-vector $l(x)$ where

$$l_i(x) = f_i - x_i + 1 \quad i = 1, \ldots, n$$

and

$$l_0(x) = 1 .$$

An n-dimensional simplex with vertices $(y^0, y^1, \ldots, y^n)$ which satisfies the equation

$$\sum_{i=0}^{n} \lambda_i l(y^i) = e ,$$

where $e = (1, \ldots, 1)$, with nonnegative λ_i's, then gives a good approximation to a fixed point x^* of the mapping $f(x)$.

4. Development and Applications of Simplicial Algorithms

Historically fixed points have played an important role. Poincaré seems to have been the first to apply fixed point results to state existence theorems [73]. After Brouwer's proof of the fixed point theorem now named after him [10], a large number of other fixed point theorems have been developed. These fixed point theorems were then used to show that complicated nonlinear problems had a solution, see, e.g., Istratescu [46] or Dugundji and Granas [25]. It was a theoretical approach with no constructive method to back it up. It is well known that Brouwer later in his life distanced himself from his most famous theorem. In [11] Brouwer shows that there is no way of determining a fixed point as a consequence of the invalidity of the Bolzano–Weierstrass theorem in intuitionism. It is interesting that he replaced his original theorem by a theorem which states

that we can only find an approximate fixed point. This is the very thing fixed point algorithms allow us to do.

Debreu introduced fixed point theorems in his book "Theory of Value" [22]. Fixed point theorems, especially Brouwer's theorem and Kakutani's theorem [48], were used to show the existence of solutions or the existence of equilibrium prices in economic models. Because of the importance of such models the race was then on not only to be able to theoretically show that an equilibrium price exists, but also to actually compute this price. In 1967 Scarf [83] succeeded in giving the first constructive proof of Brouwer's fixed point theorem using 'primitive sets', a concept related to concepts used in linear programming. See also [84]. Various other constructive proofs of Brouwer's fixed point theorem using more traditional mathematical methods followed in quick succession, see, e.g., Todd [93]. Other algorithms are, e.g., Eaves [26], Saigal [81], Allgower and Keller [2], Wright [98], etc.

A new way of attaching labels to vertices of simplices was introduced by Merrill in his Ph.D. thesis [66]. A further improvement of first generation simplicial algorithms was introduced by van der Laan and Talman see [60, 92]. They introduced algorithms which could start anywhere on the simplex and not just at a vertex. For another approach to the restart problem see Tuy [96].

The next improvement developed was grid refinement. In order to obtain better accuracy without having to restart the whole algorithm, new algorithms were developed which allowed to use the results obtained on coarse grids and refine the solution further on finer grids. Kuhn and MacKinnon developed the sandwich method [56]. Van der Heiden [44] showed that Scarf's 'primitive sets' could be extended to a grid refinement algorithm. For others see, e.g. [93]. See also Kojima and Yamamoto [51].

An important practical part of simplicial algorithms are the actual details of triangulations. For algorithms which start on a coarse grid and then obtain better and better approximations on finer grids, a concise form of the pivoting rules is needed. Pivoting rules for various triangulations are, e.g., given in Todd's book [93]. Furthermore, see, e.g., Kojima [50], and Eaves' book [27].

As an example we give one of the simplest pivoting rules. Simplices are described by one starting vertex and a number of vectors which are added in a certain order to this starting vertex. Each endpoint of such a vector is a vertex of the simplex. This ordered way of adding vectors allows to give a number to each vertex of the simplex. The algorithm requires us to drop vertex number i from a simplex σ when, e.g., the integer label obtained from the labelling function is duplicated. What we then need is to obtain a new vertex replacing the dropped vertex, so that we obtain a new simplex τ, adjacent to the simplex σ. A pivoting table is a simple way of expressing this in concise form. We give the simplest pivoting table for $\mathbb{R}^n$ with triangulation $K1$.

The simplex σ is described by a starting vertex y^0 and has unit-vectors $u^{\pi(j)}$ added in the order described by a permutation of $1, \ldots, n$, which we call π. The

new simplex τ has starting vertex z^0 and the unit-vectors $u^{\rho(j)}$ are then added in the order determined by the permutation ρ. The pivoting table allows us to find the new starting vertex z^0 and the new permutation ρ, depending on the dropped vertex i and using the old starting vector y^0 and the old permutation π. The number of subdivisions in the unit-vectors u is m.

	$z^0 =$	$\rho =$
$i = 0$	$y^0 + \frac{1}{m} u^{\pi(1)}$	$(\pi(2), \ldots, \pi(n), \pi(1))$
$0 < i < n$	y^0	$(\pi(1), \ldots, \pi(i+1), \pi(i), \ldots, \pi(n))$
$i = n$	$y^0 - \frac{1}{m} u^{\pi(n)}$	$(\pi(n), \pi(1), \ldots, \pi(n-1))$

Another practical aspect of simplicial algorithms is concerned with minimizing the number of simplices in an n-dim cube. It is straightforward to subdivide an n-dim cube into $n!$ simplices. Mara [64] has shown that it is possible to subdivide a 3-dim cube into 5 simplices instead of $3! = 6$ simplices. A smaller number of simplices in a cube may lead to a faster algorithm, because the number of simplices which has to be traversed may be smaller. To find the minimal triangulation of n-dim cubes is obviously a combinatorial problem of some importance. For further investigations see, e.g., Cottle [16], and Sallee [82]. For some recent descriptions of triangulations which allow a reduction in the computational cost of simplicial algorithms see Dang [17, 18, 20] and Dang and Talman [19]. For a further development of Dang's triangulation, see [94].

Simplicial algorithms have been used in a number of applications, see, e.g. [79, 49, 28]. The usefulness of simplicial algorithms can, e.g., be seen in an economic application modelling the effect of the U.K. joining the E.E.C. [67]. An application in connection with a U.S. tax model is given in [85]. An example of telephone pricing is given in [23]. Allgower used simplicial algorithms for solving discretized versions of nonlinear differential equations [3]. See also the excellent survey [4] and the recent book [5] which includes computer programs.

A wide range of applications in the engineering sciences is given in Watson [97], but he uses mainly methods which we will briefly describe in the next section.

5. Algorithms Based on Nonsimplicial Techniques

A number of algorithms have been published based on developments in differential topology. The most influential background books are Milnor [68] and Guillemin and Pollack [42]. Chow *et al.* [15] published a proof of Brouwer's fixed point theorem based on material from differential topology. The theorem makes use of Sard's theorem and the algorithm works with probability one. See also Alexander and Yorke [1]. A recent very detailed exposition, including FOR-

TRAN programs, of these algorithms can be found in Morgan's book [69]. Algorithms based on differential topology have a number of theoretical flaws when one takes into account that these algorithms are implemented on digital computers.

First of all, these algorithms work except on sets of measure zero. It is well known that a finite set of numbers has measure zero. The machine numbers, i.e., the numbers available on a digital computer, represent a set of measure zero. Nevertheless, nobody has yet published a counterexample, where the machine numbers are the set of measure zero, where such an algorithm does not work. In practical examples this flaw did not seem to have any noticeable influence on the working of such algorithms.

The other major flaw of these algorithms is the fact that theoretically one follows a continuous path, but on a digital computer one has only discrete points. One hopes that the discrete points the algorithm actually follows represent points of a continuous path. In applications with more than one solution it has actually been observed that the algorithm jumps from one solution path to another (i.e., the discrete set of points belonging to one solution path are left and the discrete set of points belonging to another solution path are followed). This happens where solution paths are close together, i.e., the step size of the algorithm is of the same order as the distance between two solution paths. Morgan, e.g., describes this unhelpful feature [69, p. 9]. Various measures have been proposed to remedy this situation, but all these measures are only an ad hoc solution to a mismatch of a body of continuous theory, namely differential topology, and implementation on a discrete machine, namely a digital computer.

For a selection of applications of such algorithms see, e.g. [15, 97]. Applications to eigenvalues and eigenvectors are given in [63].

Another example of the influence of modern mathematical developments on algorithms for the solution of systems of nonlinear equations is the book by Rheinboldt on parameterized nonlinear equations [77]. In this case differential geometry is used as theoretical background and continuation methods are used to trade one-dim solution submanifolds. A FORTRAN package is included.

6. Kuhn's Algorithm for Finding All Zeros of a Polynomial

One of the most significant developments in solving equations with more than one solution is Kuhn's method for finding all roots for a single polynomial. This method which is based on topological principles and on a generalized version of Sperner's lemma can only be described as ingenious. From the coefficients of a polynomial of degree n we can *a priori* determine the size and mesh of a triangulated region which will contain all solutions.

In Kuhn's notation we have a polynomial

$$f(z) = z^n + a_1 z^{n-1} + \cdots + a_n$$

with in general complex coefficients a_k. The size of the square region in the complex plane which we have to triangulate ranges from $[-R, +R]$ on the real axis and from $[-iR, +iR]$ on the imaginary axis. R is given by

$$R = 48 \max_k |a_k| + 4 .$$

In order to triangulate the square region we have to subdivide the interval $[0, R]$. The minimum number of subdivisions M we have to have on $[0, R]$ is given by

$$M \geq \frac{1}{2}(n+2) ,$$

where n is the degree of the polynomial. The labelling function will then lead to n starting segments with appropriate labels (e.g., labels 01), on the boundary of this region. Pivoting steps will lead us to n completely labelled triangles (e.g., labelled 012) inside the region representing n approximate solutions to the roots of the polynomial [55].

This method not only represents a new proof of the fundamental theorem of algebra, but also includes its own convergence proof, when regarded as a grid refinement algorithm (i.e., when we have layers of finer and finer grids connected by a suitable triangulation). The method has been developed in 3 papers [55, 57, 58] including computational complexity and *a priori* error estimates for the approximate solutions obtained. For other complexity results see [75].

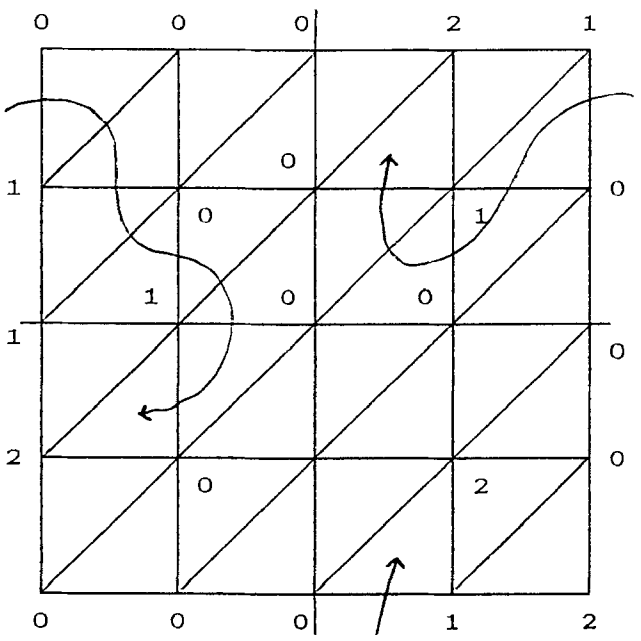

Fig. 2. Paths for a polynomial of degree 3 $f(z) = z^3 + (0.3 + 1.1i) = 0$ on part of the triangulated region.

Error estimates: If we have found a completely labelled triangle and we have used the labelling function given by Kuhn, then a root of the polynomial $f(z)$ is closer to this triangle than $(6n\delta)/\pi$, where δ is the mesh size of the triangle [57].

In [58] a tighter bound is given for polynomials of degree $n > 1$: There is a root of $f(z)$ closer than $3/4n\delta$ to the completely labelled triangle. In Section 12 we will generalize above error estimates to arbitrary labelling functions.

A computational complexity result is also available [58]: In order to evaluate a root $\bar{z}_j$ of $f(z)$ with accuracy η, the number of function evaluations of $f(z)$ is not greater than $O(n^3 \log_2(n/\eta))$.

Kuhn's method has extremely desirable properties. It is a method based on sound mathematical principles. Topological properties are used, i.e., properties invariant under deformations, which is not only important as a technique for tracing approximations of roots, but also important when we take the inevitable rounding errors on computers into account. Pivoting steps provide a secure path to the approximate solution (no "jumping" from one solution path to another solution path is possible). The number of solutions is known *a priori* and we can guarantee that this number of solutions will actually be found (no "probability" of failure associated with this algorithm). The size of the region in which the algorithm operates is known *a priori* and the mesh size for the triangulation of the region is also known *a priori* (no "guessing" of starting values "sufficiently close to a solution" is required). It would be very desirable to be able to generalize Kuhn's algorithm to systems of polynomial equations and retain as many of the above mentioned desirable properties as possible. In order to do this, we first have to clean up one of the shortcomings of Kuhn's convergence proof.

For the n isolated roots of a polynomial $f(z)$ of degree n, we have n sequences z_{jk}, $j = 1, \ldots, n$, converging to the n roots $\bar{z}_1, \ldots, \bar{z}_n$, i.e.,

$$\lim_{k\to\infty} z_{jk} = \bar{z}_j, \quad j = 1, \ldots, \ldots, \ldots, n,$$

on our triangulation with finer and finer grid. In dealing with multiple roots Kuhn first perturbs the polynomial $f(z)$ by adding a term $\varepsilon > 0$, i.e., he perturbs the polynomial with multiple roots $f(z)$ to a polynomial with simple roots $f(z) + \varepsilon$. He then considers isolated roots and then lets ε tend to zero. For the roots $\bar{z}_j$ of this polynomial we have to show

$$\lim_{\varepsilon\to 0+} \lim_{k\to\infty} z_{jk}(\varepsilon) = \lim_{k\to\infty} \lim_{\varepsilon\to 0+} z_{jk}(\varepsilon) = \lim_{k\to\infty} z_{jk} = \bar{z}_j .$$

The use of properties of complex mappings suggests this approach. Nevertheless, it is now generally accepted by researchers who have considered the proof [57] that there is a gap in the proof [53, 76, 59]. If we regard the roots $\bar{z}_j$ of a polynomial as continuous functions of the coefficients a_k, then at each grid level $d = 0, 1, \ldots$ we have an approximation z_{jd}, $d = 0, 1, \ldots$. For the perturbed polynomial $f(z) + \varepsilon$ we have approximations $z_{jd}(\varepsilon)$. Therefore, we have a se-

quence of functions $z_{jd}(\varepsilon)$, $d = 0, 1, \ldots$ and in order to be able to interchange limits

$$\lim_{\varepsilon \to 0+} \lim_{d \to \infty} z_{jd}(\varepsilon) = \lim_{d \to \infty} \lim_{\varepsilon \to 0+} z_{jd}(\varepsilon),$$

we have to show uniform convergence of the sequence z_{jd}, $d = 0, 1, \ldots$, which seems to be missing in [57]. Uniform convergence seems difficult to establish in this case.

It has been reported [59] that computational results for problems involving multiple roots seem to confirm that for smaller and smaller grid sizes (grid refinement) some of the approximate roots show erratic behaviour and do not seem to converge to the actual roots of the polynomial. This is taken as confirmation that the theoretical flaw in the proof for multiple roots actually manifests itself in the algorithm when multiple roots are present.

In the next sections we will show: (a) that Kuhn's method is sound; (b) that a satisfactory convergence proof for multiple roots can be given without perturbing the original polynomial (by introducing elements from Nielsen fixed point theory, a body of research which deals with the minimum number of fixed points invariant under homotopies); and (c) we offer an explanation for the computational tests in which multiple roots do not seem to converge to the actual roots when the approximations are followed from a layer d to the next layer $d + 1$ with finer grid.

7. Some Nielsen Fixed Point Theory

In order to give a convergence proof for Kuhn's algorithm for the case of multiple roots, we will use ideas from Nielsen fixed point theory. This will allow us to avoid the difficulties connected with the interchange of limits discussed in the last section.

To illustrate the beautiful ideas behind Nielsen fixed point theory, we will first show that on a unit circle $\mathbb{S}^1$ in the complex plane $\mathbb{C}$ two polynomials, a polynomial $f(z)$ of degree d_1 and a polynomial $g(z)$ of degree d_2, have at least $|d_1 - d_2|$ coincidence points, that is $|d_1 - d_2|$ points z^* such that $f(z^*) = g(z^*)$ on $\mathbb{S}^1$. We can then specialize $g(z)$ either to z, i.e., to finding points z^* which satisfy the fixed point equation $f(z^*) = z^*$ on $\mathbb{S}^1$. Alternatively, we can specialize $g(z)$ to an arbitrary constant $a \in \mathbb{S}^1$, i.e., we can specialize to the equation $f(z) = a$ in order to find constant points z^* such that $f(z^*) = a$ on $\mathbb{S}^1$ (i.e., the argument value $\phi = a$ on $\mathbb{S}^1$). For the equation $f(z) = 0$ on $\mathbb{S}^1$ we can choose as a special case the argument value $\phi = 0$ on $\mathbb{S}^1$ and look for constant values z^* such that $f(z^*) = 0$ on $\mathbb{S}^1$. We will deal in solution classes and will show that each solution class is not empty. Then we will show that Kuhn's algorithm converges even in the case of multiple roots.

Let $\mathbb{S}^1 = \{z \in \mathbb{C} \mid |z| = 1\}$ be the unit circle in the complex plane $\mathbb{C}$. Let

$$p : \mathbb{R} \to \mathbb{S}^1$$

be the exponential map, i.e.,

$$p(\phi) = z = e^{i\phi} ,$$

with ϕ the argument of z. The argument of z is a multivalued function of z, i.e., for every map

$$f: \mathbb{S}^1 \to \mathbb{S}^1$$

we can always find argument expressions

$$\tilde{f}: \mathbb{R} \to \mathbb{R}$$

such that

$$f(e^{i\phi}) = e^{i\tilde{f}(\phi)} .$$

We can find a whole series of such argument expressions differing from each other by integer multiples of 2π. We will write $\tilde{f}_0$ for the argument expression with

$$\tilde{f}_0(\phi = 0) \in [0, 2\pi) .$$

We have $\tilde{f}_k = \tilde{f}_0 + 2k\pi$. The degree of f is d_1 and therefore the functions $\tilde{f}_k$ are such that

$$\tilde{f}_k(\phi + 2\pi) = \tilde{f}_k(\phi) + d_1 2\pi .$$

If $z = e^{i\phi}$ is a coincidence point, i.e.,

$$f(z) = g(z) ,$$

then ϕ is a coincidence point of some argument expression of f, i.e., $\tilde{g}(\phi) = \tilde{f}_k(\phi)$ for some k. On the other hand, if ϕ is a coincidence point of $\tilde{f}_k$ and q is an integer, then we can show that $\phi + q2\pi$ is a coincidence point of $\tilde{f}_l$ iff

$$l - k = q(d_1 - d_2) .$$

This follows from

$$\begin{aligned}
&\tilde{f}_l(\phi + q2\pi) \\
&= \tilde{f}_0(\phi + q2\pi) + l2\pi \\
&= \tilde{f}_k(\phi + q2\pi) - k2\pi + l2\pi \\
&= \tilde{f}_k(\phi) + d_1 q2\pi + 2\pi(l - k) \\
&= \tilde{g}(\phi) + d_1 q2\pi + 2\pi(l - k) \\
&= \tilde{g}(\phi) + d_2 q2\pi + d_1 q2\pi + 2\pi(l - k) - d_2 q2\pi \\
&= \tilde{g}(\phi + q2\pi) + 2\pi\{q(d_1 - d_2) - (k - l)\} .
\end{aligned}$$

Therefore, if

$$l \not\equiv k \bmod(d_1 - d_2) ,$$

then a coincidence point of $\tilde{f}_k$ and a coincidence point of $\tilde{f}_l$ can never correspond to the same coincidence point.

We see that the argument expressions fall into equivalence classes by the relation

$$\tilde{f}_k \sim \tilde{f}_l \quad \text{iff } l \not\equiv k \bmod(d_1 - d_2),$$

and the coincidence points split into $|d_1 - d_2|$ classes. Two coincidence points of $f(z)$ and $g(z)$ are in the same class iff they come from coincidence points of the same argument expression.

To show that a map f of degree d_1 and a map g of degree d_2 have at least $|d_1 - d_2|$ coincidence points, we only have to show that every coincidence point class is not empty, or equivalently, that every argument expression has a coincidence point if $d_1 \neq d_2$. We have

$$\tilde{f}_k(\phi + 2\pi) - \tilde{f}_k(\phi) = 2\pi d_1$$

and

$$\tilde{g}(\phi + 2\pi) - \tilde{g}(\phi) = 2\pi d_2,$$

and therefore $\tilde{g}(\phi) - \tilde{f}_k(\phi)$ takes on different signs when ϕ approaches $+\infty$ or $-\infty$. That means that $\tilde{f}_k$ has at least one coincidence point.

Above example is a generalization of the approach taken by Nielsen in his original paper [70], in which he determined the minimum number of fixed points of a continuous mapping of a torus into itself. In general Nielsen fixed point theory determines equivalence classes. For an introduction to Nielsen fixed point theory see, e.g., Brown [12] or Jiang [47].

Fixed point classes are defined as follows.

DEFINITION. Two fixed points x_0 and $x_1 \in X$ of a mapping $f: X \to X$ belong to the same fixed point class iff there is a path c from x_0 to x_1 such that c is homotopic to $f \circ c$, i.e., if it is possible to define a deformation of c into $f \circ c$.

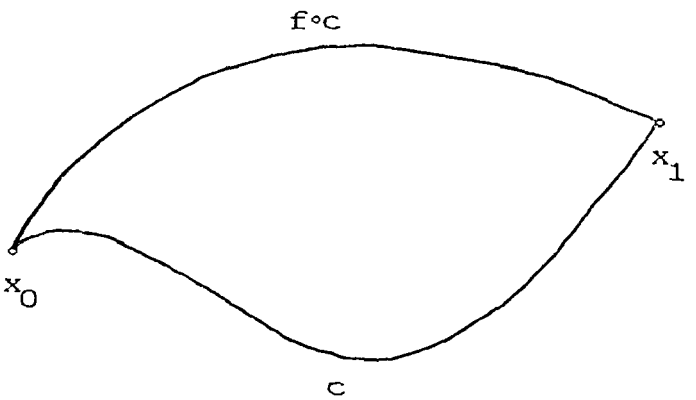

Fig. 3. Illustration of homotopic paths c and $f \circ c$ used in Nielsen fixed point theory.

Nielsen fixed point theory tries to determine the minimum number of such equivalence classes. The importance of Nielsen fixed point theory in the context of homotopy algorithms lies in the fact that the Nielsen number $N(f)$, i.e., the number of fixed point classes, is a homotopy-invariant. That means, if a map has $N(f)$ fixed points, then any map homotopic to f has at least $N(f)$ fixed points. We will use this fact to establish the covergence of Kuhn's algorithm in the case of multiple roots (simple roots are adequately dealt with in Kuhn's paper [57]).

8. Correction to Kuhn's Convergence Proof

First we show that a polynomial of degree n behaves like z^n for large $|z|$. Given the polynomial

$$f(z) = z^n + a_1 z^{n-1} + \cdots + a_n ,$$

we construct the following homotopy

$$f(z, t) = tf(z) + [1 - t]z^n = z^n + t[a_1 z^{n-1} + \cdots + a_n] ,$$

with $t \in [0, 1]$.

We obtain

$$\frac{f(z, t)}{z^n} = 1 + t[\underbrace{\ldots\ldots\ldots\ldots\ldots}_{\to 0 \text{ as } |z| \to \infty}] ,$$

i.e., the polynomial shows the behaviour of z^n for large $|z|$ (or on a large circle S^1).

From our example dealing with coincidence points on $\mathbb{S}^1$ in Section 7, we know that z^n has n solution classes, i.e., we will have n solution points for $z^n = 0$ on the unit circle $\mathbb{S}^1$ (look for points with argument value $\phi = 0$ on $\mathbb{S}^1$).

By using an appropriate labelling function and piecewise linear approximations we replace the large circle S^1 by a piecewise linear approximation and each solution point on S^1 by a 1-dimensional simplex labelled, e.g., 01. We then trace the path of these 01 labelled simplices by pivoting steps until we obtain a simplex labelled 012. Pivoting can be considered in terms of homotopies. One pivoting step can be considered to be a collapse across a 2-dimensional simplex.

Because collapsing can be used to establish equivalent paths (simplicial homotopies) and therefore a connection between pivoting and homotopy, we will give the definition.

DEFINITION. Let K be a simplicial complex. An n-simplex σ of K is said to have a free face τ, if τ is an $(n-1)$-face of σ but is a face of no other n-simplex of K. If σ has a free face, then $K - \sigma - \tau$ is a subcomplex of K.

The process of passing from K to $K - \sigma - \tau$ is called an elementary collapse. If L is a subcomplex of K, K is said to collapse to L, written $K \searrow L$, if L can be obtained from K by a sequence of elementary collapses.

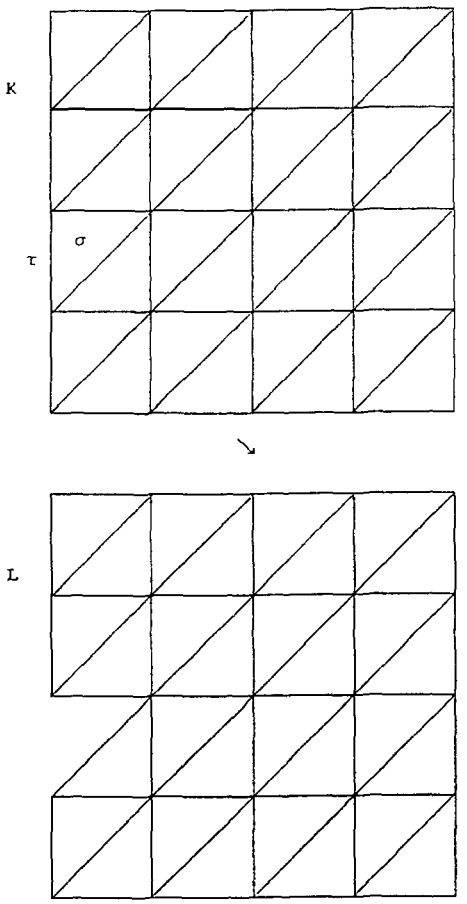

Fig. 4. Illustration of a collapse of K onto L.

For more details on collapsing see, e.g., Zeeman [100, chapter 3], Rourke and Sanderson [80], and Maunder [65].

We consider completely labelled simplices (i.e., simplices labelled 012) as approximations to the zeros of the polynomial $f(z)$ in the complex plane $\mathbb{C}$. For a polynomial of degree n we have on a large circle S^1 (i.e., on the boundary of the region) a behaviour like z^n. Each of the n solutions to $z^n = 0$ on the boundary circle S^1, i.e., $z \in S^1$, belongs to a different solution class. Replacing the large circle S^1 by a piecewise linear approximation to S^1 is a homotopy. Each of the n solutions on the boundary of S^1 is replaced by a simplex labelled 01. We can regard pivoting in order to find a simplex labelled 012 as replacing the piecewise linear approximation to the large circle S^1 by a sequence of homotopic piecewise liner circles (we consider only piecewise linear circles in $\mathbb{C}$ such that $\| f(z) \| \neq 0$ on such a piecewise linear circle and therefore we retain the number of solution classes according to Nielsen).

We now consider the piecewise linear path generated by the vertices which are, e.g., labelled 0. This piecewise linear path can be regarded to be a piecewise linear approximation to the original continuous path from the boundary to the actual zero. The original continuous path leads from the boundary to a zero, the piecewise linear approximation leads from the piecewise linear boundary (i.e., from a simplex labelled 01) to a piecewise linear approximation of the zero (i.e., a simplex labelled 012). So, despite the fact that we are on a triangulated region with discrete vertices, we can talk about continuous paths from the boundary to the approximate solutions.

We now consider grid refinement (and therefore convergence to the solutions of the polynomial $f(z) = 0$, $z \in \mathbb{C}$, for finer and finer grids). First we use the simplices labelled 012 obtained in the starting layer $d = 0$. We assume we have a piecewise linear circle S_{012} in layer $d = 0$, which includes all the simplices labelled 012 obtained by pivoting from the large circle S^1 on the boundary. We further assume that the faces labelled 01 of the simplices labelled 012 are part of this piecewise linear circle. We now pivot with the simplices labelled 012 until we are in a layer which gives us the required accuracy (i.e., we pivot from layer $d = 0$ to layer $d = k$). Because pivoting changes one vertex of a simplex at a time, we can always deform the circle S_{012} into another circle homotopic to S_{012}, and which includes the new vertex if the new vertex is labelled 0 or 1. Proceeding in this manner until layer $d = k$, we will have the same count of smaller simplices (labelled 012) as in the starting layer $d = 0$ (but we have a more accurate approximation to the zeros of the polynomial $f(z)$) and a new circle homotopic to S_{012} (therefore according to Nielsen retaining the number of solutions). Using collapsing we can then deform the circle S_{012} in such a way, that the circle in its entirety is in layer $d = k$ (we can do this without changing the count of simplices labelled 01, if we use free edges labelled jj with j either 0, 1, or 2, and collapse a 2-dimensional simplex only onto an edge again labelled jj).

Another way of looking at the process of pivoting from layer $d = 0$ to layer $d = k$, is to regard going to a finer grid as replacing one piecewise linear approximation of the polynomial $f(z)$ with a finer piecewise linear approximation of $f(z)$.

If on the other hand, we go back to the boundary where we started the algorithm (i.e., layer $d = 0$), and pivot with the simplices labelled 01 on the boundary to the boundary of the region with the finer grid (i.e., we pivot restricted to the boundary from layer $d = 0$ to layer $d = k$), then this can again be regarded as a homotopy of piecewise linear circles. The number of simplices labelled 01 will be the same as before (in other words according to Nielsen the number of solution classes will be the same). If we then pivot from the boundary of the region with th finer grid (i.e., from the boundary of layer $d = k$) into the interior of that region, we will again obtain the same number of simplices labelled 012 (being a better approximation than the simplices labelled 012 in the starting layer $d = 0$).

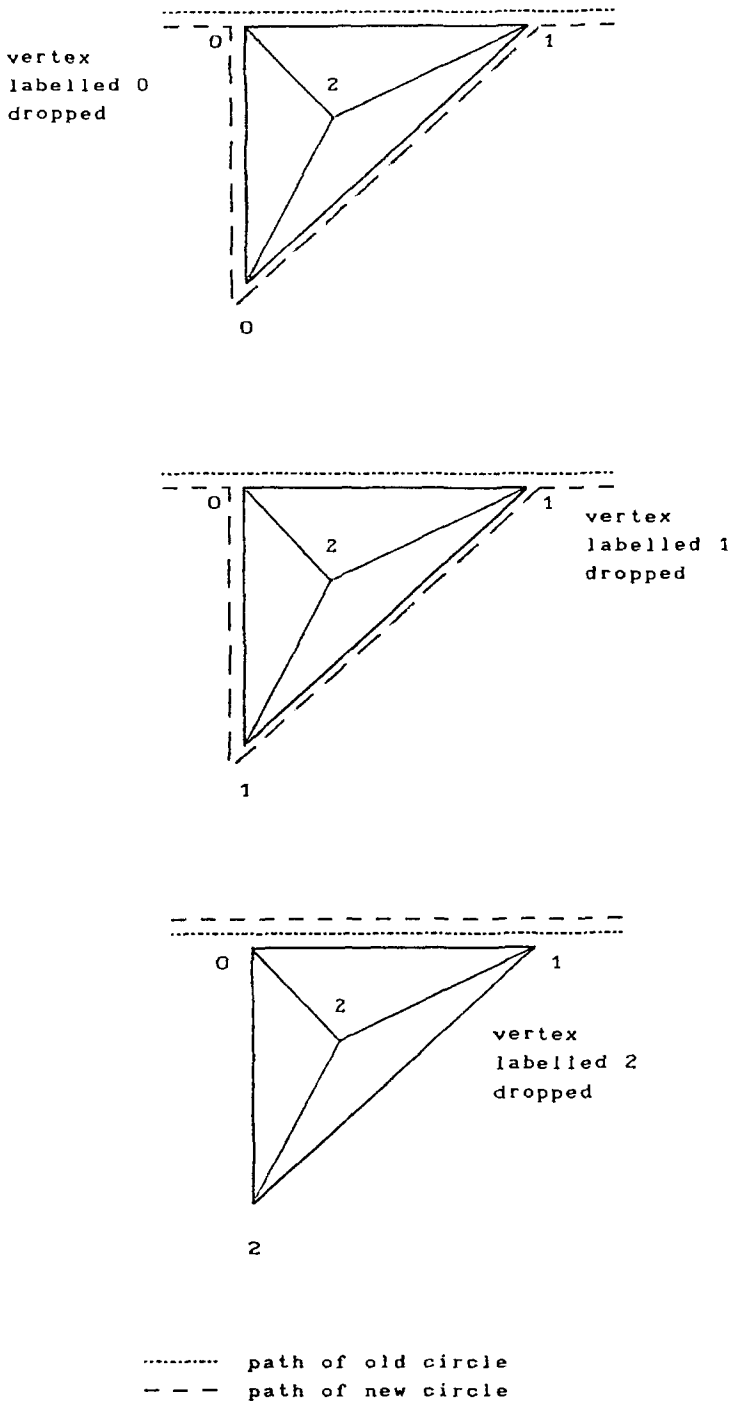

Fig. 5. Deformation of part of a piecewise linear circle during pivoting for grid refinement.

The two approximations, i.e., two simplices obtained by (i) first pivoting in the starting layer $d = 0$ from a simplex labelled 01 on the boundary to a simplex labelled 012 and then using grid refinement; and (ii) first pivoting from a simplex labelled 01 from the boundary of the starting layer $d = 0$ to a simplex labelled 01 on the boundary of the layer $d = k$ with finer grid and then into the inside of the layer $d = k$ until we find a simplex labelled 012, are in general not the same. Nevertheless, these two solutions (better the two 01 faces of the two simplices labelled 012) can be obtained from each other by homotopy of circles (homotopies of piecewise linear approximations of circles) and are therefore in the same solution class!

Further grid refinement will give a better and better approximation and because a solution from one solution class can never be deformed into a solution from another solution class (a result of Nielsen fixed point theory) we have our convergence proof.

This proof holds for simple and multiple roots. For small enough mesh sizes δ we can isolate roots in the following sense. If the actual roots (simple or multiple ones) of the polynomial $f(z)$ are more than $2(6n\delta)/\pi + 2\delta$ apart, then we can draw a piecewise linear circle around each simple root or a piecewise linear circle

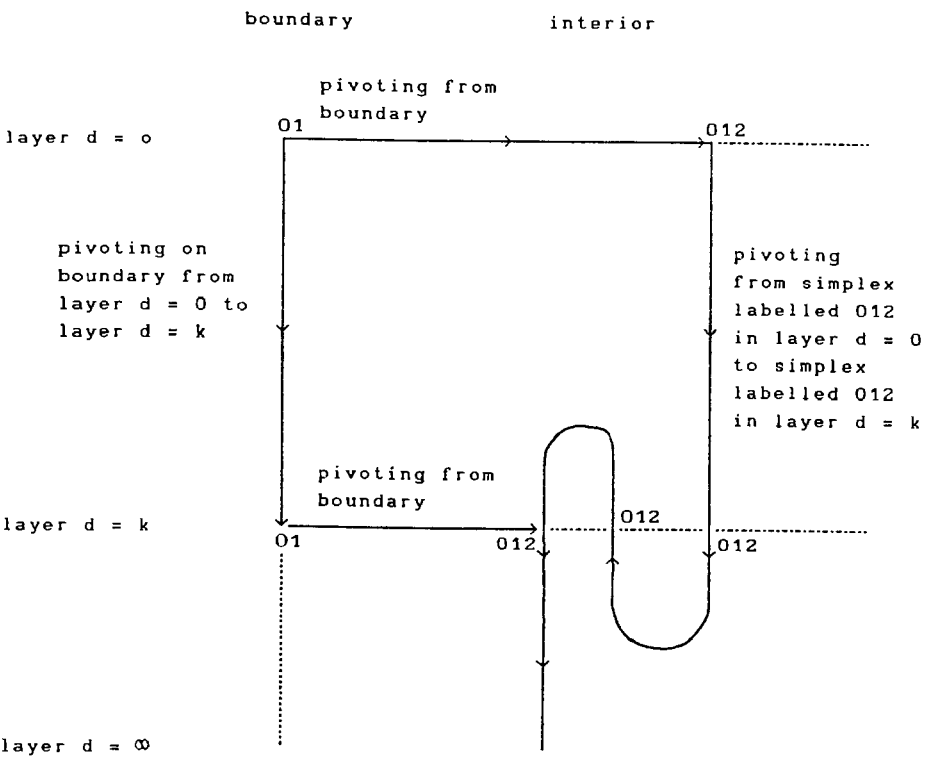

Fig. 6. Schematic representation of homotopy equivalence of approximate solutions in different layers.

around multiple roots. Using piecewise linear approximations we will have a single path of simplices labelled 01 from the boundary (i.e., the large circle S^1) of that layer to the piecewise linear circle surrounding a simple root, or we will have more than one path of simplices labelled 01 (the multiplicity of the multiple root number of paths) from the boundary (i.e., the large circle S^1) to the piecewise linear circle surrounding the multiple root. The simplices labelled 012 obtained by grid refinement, i.e., by pivoting from a simplex labelled 012 in the starting layer $d = 0$ to the refined layer $d = k$, are inside the above mentioned circles surrounding the roots [by virtue of Nielsen fixed point theory (we preserve solution classes and the number of solution classes), by the generalized Sperner lemma (the count of simplices labelled 01 on the boundary is the same as the count of simplices 012 in the interior), and the error estimate $(6n\delta)/\pi$, which holds for all roots (simple or multiple roots) and draws the simplices labelled 012 closer to a root as $\delta \to 0$].

Because in our proof we are always considering homotopies of circles (or homotopies of piecewise linear circles), we are able to show that Kuhn's homotopy algorithm will approximate n different roots of the polynomial of degree n (by virtue of the n solution classes). We have therefore clarified the convergence process and given a convergence proof without perturbing the original polynomial!

9. Numerical Observations for Kuhn's Algorithm

What we now have to explain is the computational experience in which multiple roots do not seem to converge to the actual roots. Kuhn [55] uses a labelling function which involves $\arctan(x/y)$. For small values of x (real coordinate of the point z in the complex plane) or y (imaginary coordinate of the point z in the complex plane) $\arctan(x/y)$ is very sensitive to rounding errors. Our own computational tests have led to the observation that different implementations of this labelling function lead to different numerical behaviour.

Use of higher accuracy for the representation of numbers (e.g., use of double precision instead of single precision in FORTRAN, or, e.g., of the use of the type 'double' instead of 'real' in Turbo PASCAL) leads to different labels near the solution and so to different paths near the solutions.

Another difference we found in our computational tests relates to the way the quotient (x/y) is evaluated. Separate evaluation of $u = (x/y)$ and then using $\arctan(u)$ leads to results which are much closer to the true solutions than using $\arctan(x/y)$ without evaluating the quotient (x/y) separately.

Nevertheless, we found that in all cases the solutions were distributed correctly in the sense that within a distance of $(6n\delta)/\pi$ from each of the n known solutions (including the multiple solutions) we had the correct number of approximate solutions.

Changes in labels and paths are attributable to the representation of arctan in computers. It is difficult to find out how $\arctan(x/y)$ is actually represented in a particular implementation, but the series expansion of $\arctan(x/y)$ including terms of the form $(x/y)^{2j+1}$ or $(y/x)^{2j+1}$ can be very sensitive to small errors in x or y.

All this indicates that Kuhn's algorithm is sound.

10. Measures of Efficiency and a Paradox

A large number of different triangulations for simplicial algorithms has been published. A need was felt to be able to compare the 'efficiency' of various triangulations. Measures of efficiency were defined and applied to triangulations. As a measure of efficiency one could, e.g., count the number of simplices traversed by a straight line. For a good introduction see, e.g., Todd [93]. Whereas these measures of efficiency of a triangulation (e.g., directional density, horizontal directional density and vertical directional density for algorithms with grid refinement, etc.) gave some indication of the work required in simplicial algorithms, in general these measures do not catch the advantages gained by grid refinement algorithms (or similar schemes).

To illustrate this let us take a paricular algorithm, namely Kuhn's algorithm for finding roots of polynomials. Intuitively it is obvious that when we start with a coarse grid and then refine the grid, the work required is less than by using just one layer of fine grid (for solutions required to have a certain accuracy η). In the first case we take a small number of large steps on the coarse grid (which bring us quickly close to a solution) and then smaller and smaller steps during grid refinement (which brings us still closer to a solution). In the second case we will have a very large number of small steps on a fine grid (and we will make only slow progress towards a solution). For a polynomial of degree n using Kuhn's algorithm and only one layer (i.e., one grid size) the number of function evaluations in the worst possible case is

$$O(n^2) .$$

If we use the grid refinement version of the algorithm, then the number of function evaluations in the worst possible case is

$$O(n^3 \log_2(n/\eta)) ,$$

see [58].

If we want to obtain solutions with accuracy η, then above estimates make the intuitively better algorithm (i.e., grid refinement) appear to be the inferior choice.

Can one construct measures of efficiency which resolve this paradox? This is an open question.

11. Systems of Nonlinear Equations and Systems of Polynomial Equations

The success of the constructive version of Brouwer's fixed point theorem and the liberation from the shackles of contraction mappings led to more ambitious projects. The new aim was to find not just one solution but 'all' solutions of systems of nonlinear equations.

In 1976 Fisher *et al.* [30] and in 1980 Gould [40] investigated the solution of a system of nonlinear equations

$$f(x) = 0 ,$$

where

$$f : \mathbb{R}^n \to \mathbb{R}^n ,$$

by simplical methods. They found that using simplicial techniques is not a guarantee for convergence. They gave examples where a simplicial algorithm always diverges and they found that reordering the equations and a change in labelling function can change convergence.

Looking back, this state of affairs is not surprising. Following a path of simplices is not a guarantee for convergence to a solution, unless theoretical convergence results are available. They developed theoretical conditions for convergence and for problems satisfying these conditions convergence could be guaranteed. At this stage only one solution was involved. For the algorithm the product space

$$\mathbb{R}^n \times [0, 1]$$

is triangulated. The algorithm starts in layer

$$\mathbb{R}^n \times \{0\}$$

with an appropriately labelled simplex, and the solution simplex is found in layer

$$\mathbb{R}^n \times \{1\} .$$

Charnes *et al.* [14] consider a system of n nonlinear equations in n unknowns

$$f(x) = a ,$$

where

$$f : \mathbb{R}^n \to \mathbb{R}^n .$$

Using a homotopy

$$H(x, t) = (1 - t)f^0(x) + t[f(x) - a] ,$$

with

$$t \in [0, 1] ,$$

and $f^0(x)$ a linear function with known solution

$$f^0(x) = 0 .$$

We have again two layers $\mathbb{R}^n \times \{0\}$ and $\mathbb{R}^n \times \{1\}$. The triangulated region is $\mathbb{R}^n \times [0, 1]$. The homotopy can be considered as a map

$$H: \mathbb{R}^n \times [0, 1] \to \mathbb{R}^n .$$

A piecewise linear solution path is then traced from layer $\mathbb{R}^n \times \{0\}$ to layer $\mathbb{R}^n \times \{1\}$. Conditions for convergence are given. Again only one solution is involved.

In 1979 the emphasis changed from finding one solution to finding 'all' solutions. What is meant by 'all' solutions is not investigated theoretically and the general form of solution sets is not investigated either. Methods are given for systems which are dominated by polynomials.

E.g., Garcia and Zangwill in [37] consider the following system

$$z_j^{s_j} + P_j(z) = 0 , \quad j = 1, 2, \ldots, n ,$$

where

s_j a positive integer,

$P_j: \mathbb{C}^n \to \mathbb{C}$, for $j = 1, \ldots, n$,

with $\mathbb{C}$ the complex plane and $\mathbb{C}^n$ the n-dimensional complex space, and $z \in \mathbb{C}^n$.

The following dominance condition is imposed

$$\left\| \frac{P_j^{(z)}}{z_j^{s_j}} \right\| \to 0 \quad \text{as } \|z_j\| \to \infty ,$$

therefore $z_j^{s_j}$ dominates the expression $P_j(z)$.

The homotopy

$$H: \mathbb{C}^n \times [0, 1] \to \mathbb{C}^n$$

has the form

$$H_j(z, t) = z_j^{s_j} + t[P_j(z) + 1] - 1 ,$$
$$j = 1, \ldots, n ,$$
$$t \in [0, 1] .$$

We trace the solutions of

$$H(z, t) = 0$$

from $t = 0$, i.e.,

$$z_j^{s_j} - 1 = 0 ,$$
$$j = 1, \ldots, n ,$$

to $t = 1$.

The use of the dominating terms $z_j^{s_j}$ has the effect to make all solutions into zero-dimensional polynomial ideals, i.e., points.

The total number of solutions we will obtain is

$$\prod_{j=1}^{n} s_j .$$

In a further paper [38] Garcia and Zangwill extend their methods to systems of the form

$$P_j(z) = 0 , \quad j = 1, \ldots , n ,$$

where

$$P_j : \mathbb{C}^n \to \mathbb{C} , \quad \text{for } j = 1, \ldots , n, \quad \text{and } z \in \mathbb{C}^n .$$

For polynomials in n variables the following dominating condition is imposed

$$\left\| \frac{P_j(z)}{z_{j-1}^{s_j}} \right\| \to 0 \quad \text{as } \|z_j\| \to \infty ,$$

i.e., $Q_j(z) = z_j^{s_j} - 1$ dominates $P_j(z)$.

The following homotopy

$$H : \mathbb{C}^n \times [0, 1] \to \mathbb{C}^n$$

is set up

$$H_j(z, t) = [1 - t]Q_j(z) + tP_j(z) ,$$
$$j = 1 , \ldots , n ,$$
$$t \in [0, 1] .$$

As before we trace solution paths from $t = 0$ to $t = 1$. In their paper they discuss path following by differential equations and simplicial pivoting. A regularity condition is imposed on the Jacobian in order to make sure that there is only a finite number of solutions. Because it is thought that one can only cope with a finite solution set, everything is done to ensure that the solution set consists only of a finite number of points.

In Garcia and Zangwill [39] the path following approach for more than one solution is further specialized to polynomial systems in more than one variable. By now it has become clear that polynomial systems in more than one variable are systems which are much easier to control than a general system of nonlinear equations. They use various dominating terms, which make sure that the solution set are zero-dimensional polynomial ideals, i.e., points.

Kojima and Mizuno [52] take up polynomial systems of the form

$$z_j^{s_j} + P_j(z) = 0 , \quad j = 1, 2, \ldots , n ,$$

and impose a dominating condition

$$\left\| \frac{P_j(z)}{z_j^{s_j}} \right\| \to 0 \quad \text{as } \|z_j\| \to \infty .$$

A homotopy for finding the total number of solutions

$$\prod_{j=1}^{n} s_j$$

is discussed. For the actual path following grid refinement is used. For the first time we have a paper which includes estimates for grid sizes for polynomial systems. For general nonlinear systems such estimates for grid sizes, which are important for the actual implementation of the algorithm, seem to be very difficult to obtain.

We have given a selection of papers dealing with systems of nonlinear equations and we have traced the development of algorithms for systems with more than one solution. The work of these researchers has highlighted a number of questions arising in connection with the solution of systems of nonlinear equations. What are 'all' solutions? Can we only deal with zero-dimensional solution sets, i.e., points as solutions? We know that general solution sets are much more complicated (as an example consider, e.g., the intersection of two cylinders). What can we say about the homotopy invariance of the number of solutions? (e.g., in $\mathbb{R}^n$ or in $\mathbb{C}^n$). When do paths diverge to infinity?

The developments outlined in this section, important as they are, do not have the same clean theoretical background as we have seen, e.g., in Sperner's lemma or Brouwer's fixed point theorem. Some authors use Bezout's theorem [7] to determine the number of solutions of polynomial systems. Bezout's theorem seems to provide a theoretical justification for the use of dominating terms, but Bezout's theorem is material that does not quite fit into a homotopy context. Bezout's theorem is from algebraic geometry, which is not concerned with homotopies. The main objection to the use of Bezout's theorem on $\mathbb{R}^n$ (or $\mathbb{C}^n$) is a homotopy result from Nielsen fixed point theory, which says that on $\mathbb{R}^n$ (or on $\mathbb{C}^n$) the number of fixed point classes is ≤ 1 (this result holds in general for simply connected spaces). Therefore, whatever the number of intersections on $\mathbb{R}^n$ (or on $\mathbb{C}^n$) obtained by Bezout's theorem, these intersections (or solutions) can then be deformed into ≤ 1 solution classes.

If we want to maintain a homotopy approach and at the same time work with more than one solution, then $\mathbb{R}^n$ (or $\mathbb{C}^n$) can immediately be ruled out as a suitable space. In order to conserve the homotopy spirit, we have to look for other spaces which allow an arbitrary number of solution classes. In the next section we will see that there are such spaces and we can work in such spaces in a natural way.

12. Further Developments of Methods for Polynomial Systems

If we consider the solution of general systems of nonlinear equations on compu-

ters we will not get very far unless we take the finiteness of processes on computers into account. This means in particular that we will only be able to find a finite number of solutions however sophisticated our methods. Functions (e.g., $\sin(x)$, etc.) are always approximated on computers. These approximations usually take numerical accuracy into account, but violate topological properties. A typical approximation is by a suitable polynomial with a finite number of terms. Such polynomial approximations can give a good numerical approximation in a given region but obviously such approximations cannot reproduce the fact that a function has, e.g., an infinite number of different zeros. Polynomials are not only suitable for approximating functions, but are also extremely suitable for our considerations. We will therefore assume that all functions in our system of nonlinear equations have been replaced by polynomial approximations. The Stone–Weierstrass approximation theorem states that any continuous function

$$F: \mathbb{C}^n \to \mathbb{C},$$

from a suitable subset of the n-dimensional complex space $\mathbb{C}^n$ into $\mathbb{C}$, can be uniformly approximated by polynomials in z_i and the complex conjugate of z_i, $i = 1, \ldots, n$. Using Nielsen fixed point theory we are then able to make *a priori* statements about the number of solution classes of such a system. Furthermore, we are able to develop globally convergent algorithms which will find an approximation to a representative from each solution class with a given accuracy. We will also be able to give an *a priori* estimate for the size of the region to be triangulated and the mesh size. In addition, we will be able to give error estimates and complexity results for this algorithm.

One of the advantages of using the Nielsen approach with solution classes is that the algorithm is not restricted to systems with zero-dimensional polynomial ideals as solution sets, i.e., points as solution sets. Furthermore, using the equivalence of solutions (as defined in Section 7) allows to resolve a widely discussed computational problem of nonlinear systems: How to treat solutions which are not points, but are lower dimensional manifolds. Using the Nielsen concept of solution classes this problem is resolved in an elegant manner by finding a representative of each solution class.

Let us first look at one polynomial in one variable and then generalize some of the material to systems of polynomials.

For a single polynomial $f(z)$ in one variable z we consider a map

$$f(z): \mathbb{C} \to \mathbb{C},$$

with $\mathbb{C}$ is the complex plane. Next we consider a map

$$p(z): S^1 \to S^1,$$

from a circle $S^1 = \{z \in \mathbb{C} \mid |z| = r\}$ in the complex plane $\mathbb{C}$ to itself, where

$$p(z) = \frac{f(z)}{\|f(z)\|} r.$$

From Section 8 we know that a polynomial of degree L behaves like z^L for large $|z|$. Furthermore, this means that $\|f(z)\| \neq 0$ on a large circle S^1.

Now we generalize to systems of polynomials

$$f(z): \mathbb{C}^n \to \mathbb{C}^n ,$$

in n variables $z = (z_1, z_2, \ldots, z_n) \in \mathbb{C}^n$, where

$$f(z) = (f_1(z), \quad f_2(z), \ldots, f_n(z)) .$$

Earlier papers explaining this approach are Forster [33, 34, 35]. We consider maps of the form

$$f_j(z): \mathbb{C}^n \to \mathbb{C}_j , \quad \text{for } j = 1, \ldots, n ,$$

where $\mathbb{C}^n$ is regarded as cartesian product of complex planes $\mathbb{C}_j$, i.e.,

$$\mathbb{C}^n = \mathbb{C}_1 \times \cdots \times \mathbb{C}_j \times \cdots \times \mathbb{C}_n ,$$

and

$$f_j(z_1, \ldots, z_n) = \sum_{l_{j1}=0}^{L_{j1}\max} \cdots \sum_{l_{jn}=0}^{L_{jn}\max} a^{(j)}_{l_{j1}\cdots l_{jn}} z_1^{l_{j1}} \cdots z_n^{l_{jn}} ,$$

with in general complex coefficients $a^{(i)}_{l_{j1}\cdots l_{jn}}$.

For $j = 1, \ldots, n$, $L_{j1}\max$ is the highest power of z_1 in $f_j(z_1, \ldots, z_n), \ldots, , L_{jn}\max$ is the highest power of z_n in $f_j(z_1, \ldots, z_n)$.

We assume that each equation

$$f_j(z) , \quad j = 1, \ldots, n ,$$

has a unique dominating term, i.e., there is a term such that the sum of the exponents

$$L_{j1} + L_{j2} + \cdots + L_{jn}$$

is strictly larger than the sum of the exponents

$$l_{j1} + l_{j2} + \cdots + l_{jn}$$

of any other term in that equation. Furthermore, we assume that the coefficient of the dominating term

$$z_1^{L_{j1}} \cdot \cdots \cdot z_n^{L_{jn}}$$

of equation j, $j = 1, \ldots, n$, is equal to 1, i.e., we have for the coefficient

$$a^{(j)}_{L_{j1}\cdots L_{jn}} = 1 , \quad \text{for } j = 1, \ldots, n .$$

Later we will discuss how a general polynomial which has no unique dominant term can be brought into this form.

We now investigate the behaviour of $f_j(z_1, \ldots, z_n)$ for large values of $|z_1|$, $|z_2|, \cdots, |z_n|$. We define homotopies

$$f_j(z_1, \ldots, z_n, t) = tf_j(z_1, \ldots, z_n) + [1-t]z_1^{L_{j1}} \cdots z_n^{L_{jn}},$$

for all $j = 1, \ldots, n$, and with $t \in [0, 1]$. For large $|z_1|, |z_2|, \ldots, |z_n|$ we obtain

$$\frac{f_j(z_1, \ldots, z_n, t)}{z_1^{L_{j1}} z_2^{L_{j2}} \cdots z_n^{L_{jn}}} = 1 + t[\cdots],$$

with $[\cdots] \to 0$ as $|z_1| \to \infty, |z_2| \to \infty, \ldots, |z_n| \to \infty$. The polynomial $f_j(z_1, \ldots, z_n)$ behaves for large $|z_1|, |z_2|, \ldots, |z_n|$ like

$$z_1^{L_{j1}} \cdots z_n^{L_{jn}},$$

i.e., it wraps round an n-dimensional torus

$$\mathbb{T}^n = S_1^1 \times S_2^1 \times \cdots \times S_n^1.$$

The polynomial $f_j(z_1, \ldots, z_n)$ wraps L_{j1} times round the first circle S_1^1, wraps L_{j2} times round the second circle, $S_2^1, \ldots,$ wraps L_{jn} times round the nth circle S_n^1.

This observation together with an existing result for the Nielsen number for the n-dimensional torus $\mathbb{T}^n$ allows us to select the right space for homotopy algorithms for systems of polynomial equations. Instead of considering maps f from the n-dimensional complex space $\mathbb{C}^n$ to itself, i.e.,

$$f(z): \mathbb{C}^n \to \mathbb{C}^n$$

(which has the disadvantage mentioned earlier of allowing only ≤ 1 solution classes), we consider maps p from the n-dimensional torus $\mathbb{T}^n$ to itself, i.e.,

$$p(z): \mathbb{T}^n \to \mathbb{T}^n$$

(which, as we shall see, allows any number of solution classes).

The n-dimensional torus $\mathbb{T}^n$ is a subspace of the n-dimensional complex space $\mathbb{C}^n$ and a generalization of a circle S^1 considered as subspace of the complex plane $\mathbb{C}$. We take a circle S_j^1 from each complex plane $\mathbb{C}_j$ and obtain $S_1^1 \times \cdots \times S_j^1 \times \cdots \times S_n^1 \in \mathbb{C}_1 \times \cdots \times \mathbb{C}_j \times \cdots \times \mathbb{C}_n$ or $\mathbb{T}^n \in \mathbb{C}^n$.

In particular, we have

$$p(z): \mathbb{T}^n \to \mathbb{T}^n,$$

where

$$p(z) = (p_1(z), p_2(z), \ldots, p_n(z)),$$

and the maps $p_j(z)$ from the torus $\mathbb{T}^n$ to a circle S_j^1

$$p_j(z): \mathbb{T}^n \to S_j^1,$$

with

$$p_j(z_1, \ldots, z_n) = \frac{f_j(z_1, \ldots, z_n)}{\| f_j(z_1, \ldots, z_n) \|} r_j$$

and the jth circle $S_j^1 = \{z_j \in \mathbb{C}_j \mid |z_j| = r_j\}$.

On a large torus $\mathbb{T}^n$ the polynomial f_j behaves like

$$z_1^{L_{j1}} \cdots z_n^{L_{jn}} ,$$

and therefore

$$\| f_j(z_1, \ldots, z_n) \| \neq 0$$

on this large torus $\mathbb{T}^n$.

We can then use the Nielsen number to draw conclusions about the number of zeros of the system

$$p(z): \mathbb{T}^n \to \mathbb{T}^n ,$$

and use this for our system of polynomial equations

$$f(z): \mathbb{C}^n \to \mathbb{C}^n .$$

The Nielsen number for the n-dimensional torus $\mathbb{T}^n$ has been determined in a very elegant way by Brooks *et al.* [9]. For an earlier statement of the minimum number of fixed points on an n-dimensional torus see Satz II in Hopf [45].

The minimum number N of roots for a system of polynomial equations

$$p(z): \mathbb{T}^n \to \mathbb{T}^n$$

with dominating terms

$$z_1^{L_{j1}} \cdots z_n^{L_{jn}} , \quad j = 1, \ldots, n ,$$

is given by

$$N = \left| \det \begin{bmatrix} L_{11} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & L_{1n} \\ L_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & L_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ L_{n1} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & L_{nn} \end{bmatrix} \right| .$$

All the entries of this determinant are exponents of z_j, $j = 1, \ldots, n$, and therefore integers. We can have any $N \geqslant 0$ as the number of solution classes and this is what we expect for systems of polynomial equations. The space $\mathbb{T}^n$ is not simply connected and therefore the earlier mentioned limitation for spaces like $\mathbb{R}^n$ or $\mathbb{C}^n$ (number of solution classes $\leqslant 1$) does not apply. On $\mathbb{T}^n$ we can have any number of solution classes.

We now describe the labelling function, i.e., the method used to attach labels to the vertices of our triangulated region.

We have a system of n equations

$$f_j(z): \mathbb{C}^n \to \mathbb{C}_j , \quad j = 1, \ldots, n .$$

Each equation $f_j(z)$ is a map from $\mathbb{C}^n$ into the complex plane $\mathbb{C}_j$. We subdivide each complex plane $\mathbb{C}_j$ into three sectors and indicate this subdivision into three

sectors by the argument values β_0, β_1, and β_2 (i.e., all complex planes $\mathbb{C}_j$ are subdivided into the same three sectors). We test arg f_{n-k} with the index k starting from 0 and running to $n-1$ (in that order).

We label $2(n-k)$ if $\beta_0 \leqslant \arg f_n < \beta_1$ and
$\beta_0 \leqslant \arg f_{n-1} < \beta_1$ and
. . . . . .
$\beta_0 \leqslant \arg f_{n-k+1} < \beta_1$ and
$\beta_2 \leqslant \arg f_{n-k} < \beta_0 + 2\pi$;

We label $2(n-k)-1$ if $\beta_0 \leqslant \arg f_n < \beta_1$ and
$\beta_0 \leqslant \arg f_{n-1} < \beta_1$ and
. . . . . .
$\beta_0 \leqslant \arg f_{n-k+1} < \beta_1$ and
$\beta_1 \leqslant \arg f_{n-k} < \beta_2$;

We label $2(n-k)-2$ if $f_{n-k} = 0$;

{We label 2 if $\beta_0 \leqslant \arg f_n < \beta_1$ and
$\beta_0 \leqslant \arg f_{n-1} < \beta_1$ and
. . . . . .
$\beta_0 \leqslant \arg f_2 < \beta_1$ and
$\beta_2 \leqslant \arg f_1 < \beta_0 + 2\pi$;

We label 1 if $\beta_0 \leqslant \arg f_n < \beta_1$ and
$\beta_0 \leqslant \arg f_{n-1} < \beta_1$ and
. . . . . .
$\beta_0 \leqslant \arg f_2 < \beta_1$ and
$\beta_1 \leqslant \arg f_1 < \beta_2$;}

We label 0 if $\beta_0 \leqslant \arg f_n < \beta_1$ and
$\beta_0 \leqslant \arg f_{n-1} < \beta_1$ and
. . . . . .
$\beta_0 \leqslant \arg f_2 < \beta_1$ and
$\beta_0 \leqslant \arg f_1 < \beta_1$.

This labelling function is a generalization of a labelling function given by Kuhn [55]. Each vertex of the triangulation receives a unique label. {The labelling functions for label 2 and label 1 are already included in the general case and have therefore been put into brackets}.

For the implementation of the algorithm we need an *a priori* estimate which connects the mesh size of the triangulation and the radii of the circles which make up the n-dimensional torus $\mathbb{T}^n$. For a single polynomial in one variable Kuhn [57] has established a connection between the mesh size and the radius of the region.

We use similar techniques for our systems of polynomial equations

$$f_j(z_1, \ldots, z_n), \quad j = 1, 2, \ldots, n .$$

First we establish conditions for a completely labelled simplex (i.e., a simplex labelled $0, 1, \ldots, 2n$) to exist in a certain region. What we will obtain is a vast generalization of the well known inequality due to Cauchy [13], which says that all the zeros of a polynomial are inside the circle $|z| < 1 + \max_k |a_k|$.

With μ_j the mesh size (related to equation f_j), $a_{J_j}^{(j)}$ the sum over all terms such that

$$l_{j1} + l_{j2} + \cdots + l_{jn} = J_j ,$$

i.e.,

$$a_{J_j}^{(j)} = \sum_{l_{j1}=0}^{L_{j1}\max} \cdots \sum_{l_{jn}=0}^{L_{jn}\max} a_{l_{j1}\cdots l_{jn}}^{(j)} z_1^{l_{j1}} \cdots z_n^{l_{jn}} ,$$

$$l_{j1} + \cdots + l_{jn} = J_j$$

the radius R_j (related to equation f_j)

$$R_j > \max_{J_j} |a_{J_j}^{(j)}| + 1 ,$$

a constant K_j

$$K_j = \max_{l_{j1}\cdots l_{jn}} |a_{l_{j1}\cdots l_{jn}}^{(j)}| ,$$

and α the smallest sector in the labelling function, i.e.,

$$\alpha = \min(\beta_1 - \beta_0, \beta_2 - \beta_1, \beta_0 + 2\pi - \beta_2) ,$$

we obtain the following inequality

$$\mu_j \geqslant \left[1 - \frac{\max_{J_j} |a_{J_j}^{(j)}|}{R_j - 1}\right] \frac{R_j}{[\prod_{i=1}^{n} L_{ji}\max][\sum_{i=1} L_{ji}\max] K_j} \frac{\sin \alpha}{2 \cos \frac{\alpha}{2}} .$$

If this inequality holds for $j = 1, \ldots, n$, then there are completely labelled simplices in a region $(z_1, \ldots, z_n) \in \mathbb{C}^n$ such that

$$z_1^{L_{j1}} \cdots z_n^{L_{jn}} > R^{L_{j1} + \cdots + L_{jn}} ,$$

is satisfied for all $j = 1, \ldots, n$. We have set $R > \max_j R_j$.

If we do not want completely labelled simplices in this region, then we have to establish a contradiction.

One way of obtaining a contradiction is to keep R fixed ($R \geqslant \max_j R_j$) and make the mesh size μ_j to violate at least one of the equations, i.e., make

$$\mu_j < \left[1 - \frac{\max_{J_j} |a_{J_j}^{(j)}|}{R_j - 1}\right] \frac{R_j}{[\prod_{i=1}^{n} L_{ji}\max][\sum_{i=1} L_{ji}\max] K_j} \frac{\sin \alpha}{2 \cos \frac{\alpha}{2}} \qquad \text{(A)}$$

for at least one $j = 1, \ldots, n$. If this is the case, we cannot have completely labelled simplices in the region $(z_1, \ldots, z_n) \in \mathbb{C}^n$ determined by

$$z_1^{L_{j1}} \cdots z_n^{L_{jn}} > R^{L_{j1}+\cdots+L_{jn}} .$$

For a fixed R, Equation (A) gives the mesh size μ we need for our algorithm.

Another possibility to obtain a contradiction is to keep a mesh size μ_j fixed and increase the radius R_j until Equation (A) is satisfied. Again, if this is the case, we cannot have completely labelled simplices in the region $(z_1, \ldots, z_n) \in \mathbb{C}^n$ such that

$$z_1^{L_{j1}} \cdots z_n^{L_{jn}} > R^{L_{j1}+\cdots+L_{jn}} .$$

Our next step is to describe the pivoting algorithm for our system of polynomial equations

$$f_j(z_1, \ldots, z_n) = 0 , \quad j = 1, 2, \ldots, n .$$

In order to obtain starting simplices it is convenient to work in polar coordinates

$$z_j = r_j \exp(i\phi_j) , \quad \text{for } 1 \leq j \leq n .$$

We keep the radius $r_j = R$ fixed (to simplify matters, we assume that the radius of all circles $S_1^1, S_2^1, \ldots, S_n^1$ is the same). Now we consider the n-dimensional cube

$$0 \leq \phi_j \leq 2\pi , \quad \text{for } 1 \leq j \leq n .$$

We assume we have computed the Nielsen number and it is not zero. We select a nonsingular 1×1 submatrix from the matrix of exponents (dominant terms). E.g., the exponent of z_s in equation k is $L_{ks} \neq 0$. We make this equation the last equation, i.e., equation number n, and rename the variable z_s into the variable z_n (and rename the old variable z_n into z_s). The nonzero exponent L_{ks} of the former variable z_s (now z_n) is now called L_{nn}. We then pivot from 0 to 2π along the corresponding axis, now called ϕ_n. The nonzero exponent now called L_{nn} will lead to L_{nn} one-dimensional simplices which contain the label $2n-1$ and another label.

Then we extend the nonsingular 1×1 submatrix to a nonsingular 2×2 submatrix by selecting appropriate elements from the matrix of exponents. We make the new equation the last but one equation, i.e., equation number $n-1$, and rename the new variable z_t into the variable z_{n-1} (and rename the old variable z_{n-1} into z_t). We then pivot in the plane determined by the coordinate axes now called ϕ_{n-1} and ϕ_n. For each of the one-dimensional simplices (with one label $2n-1$ and another label) obtained by the previous step we pivot from $0 \leq \phi_{n-1} \leq 2\pi$. The number of two-dimensional simplices which have labels $2n-1$, $2n-3$, and another label will be the same as the determinant of the nonsingular 2×2 submatrix.

Then we extend to a 3×3 nonsingular submatrix and rename the new variable into z_{n-2}. For each of the two-dimensional simplices (with labels $2n-1$, $2n-3$,

and another label) obtained by the previous step we pivot from $0 \leq \phi_{n-2} \leq 2\pi$. The number of three-dimensional simplices which have labels $2n-1$, $2n-3$, $2n-5$, and another label will be the same as the value of the determinant of the nonsingular 3×3 submatrix.

We continue this process until the full matrix of exponents is dealt with. We will find in the n-dimensional cube

$$0 \leq \phi_j \leq 2\pi\,, \quad j = 1, \ldots, n\,,$$

the Nielsen number of n-dimensional simplices labelled with the $n+1$ labels $2n-1$, $2n-3, \ldots, 3$, 1, and another label.

On this cube all coordinates r_j, $j = 1, \ldots, n$, are kept constant, i.e., $r_j = R = \text{const.}$, $j = 1, \ldots, n$.

Now we take one of the n-dimensional simplices already found and allow, e.g., r_1 to decrease in the first pivoting step. We pivot until we have found an $(n+1)$-dimensional simplex with an additional label not yet in our set of labels, i.e., $n+2$ different labels. All the other radii r_j, $j = 2, \ldots, n$, are kept constant, i.e., $r_j = R = \text{const.}$

Then we allow, e.g., r_2 to decrease in the first pivoting step and pivot until we have found an $(n+2)$-dimensional simplex with an additional label not yet in the set of our labels, i.e., with $n+3$ different labels. The radii r_j, $j = 3, \ldots, n$, are kept constant, i.e., $r_j = R = \text{const.}$

We continue in this manner until we have found a $2n$-dimensional simplex with $2n+1$ different labels, i.e., a completely labelled simplex with labels $0, 1, \ldots, 2n$. We go through this process with all the n-dimensional simplices we have found in our n-dimensional cube

$$0 \leq \phi_j \leq 2\pi\,, \quad \text{for } j = 1, \ldots, n\,.$$

The n-dimensional simplices obtained after applying above procedure represent approximations to zeros of our system of polynomial equations. Illustrations for two polynomials in two variables and three polynomials in three variables can be found in Forster [34].

We give an error estimate. If we have found a completely labelled $2n$-dimensional simplex and we have used the labelling function given earlier, then a root of the system of polynomials

$$f_j(z_1, \ldots, z_n) = 0\,, \quad j = 1, 2, \ldots, n\,,$$

is closer to this $2n$-dimensional simplex, than

$$\frac{\mu}{\alpha} L \max\left(\frac{\pi}{2}\,,\ \alpha\right),$$

where μ is the mesh size, α is the smallest sector in our labelling function, and L is the largest total degree of our system.

Next we give a computational complexity result. In order to compute all N (Nielsen number) roots of the system of polynomial equations

$$f_j(z_1, \ldots, z_n) = 0, \quad j = 1, 2, \ldots, n,$$

with accuracy η, the number of system evaluations is not greater than

$$O(N(2n)!(L^2)^n),$$

where N is the minimum number of roots, i.e., the Nielsen number, n is the number of variables, and L is the largest total degree of the system.

For practical applications it is probably prudent to choose a variable grid size in r_j-direction, i.e., a grid size which grows with the radii r_j, $j = 1, 2, \ldots, n$. One may then have to adapt this error estimate.

If we use grid refinement the corresponding computational complexity result is

$$O\left(N(2n+1)!(L^2)^n) \log_2 \frac{L\mu}{\eta}\right),$$

where N is the minimum number of roots, i.e., the Nielsen number, n is the number of variables, L is the largest total degree of the system, μ is the mesh size (assumed to change with radius R), and η is the required accuracy.

If the system

$$f_j(z_1, \ldots, z_n) = 0, \quad j = 1, 2, \ldots, n,$$

has one or more equations with no unique dominating term, then we supplement the system by one dummy equation $f_0(z_0)$ of the form

$$z_0 = 0.$$

We add to each equation $j = 1, \ldots, n$, a polynomial ideal, i.e., a term

$$z_0^{L_j+1} z_j^{L_j},$$

where $L_j = L_{j1} + \cdots + L_{jn}$, i.e., L_j is the largest total degree of equation j, and solve the new system

$$f_j(z_0, z_1, \ldots, z_n) = 0, \quad j = 0, 1, 2, \ldots, n.$$

The new system has

$$\prod_{j=1}^{n} L_j$$

solutions.

We make use of the following inequality for determinants of $n \times n$ matrices A with elements L_{ij}

$$\det A \leqslant \prod_{i=1}^{n} \sum_{j=1}^{n} |L_{ij}|.$$

This inequality shows that the Nielsen number of the new system is larger than or equal to the Nielsen number of our original system. The new system has the same solutions as the old system (apart from coordinate z_0, which we do not take into account in this context) and a number of additional solutions arising from the dominating terms added.

A more sophisticated method to deal with one (or more) equations with no unique dominating term is the following. Take the largest minor of the matrix of exponents of the equations which have dominating terms. Then use this $(n-1) \times (n-1)$ matrix together with the total degree of the equation with no dominating term to make it into the $n \times n$ matrix for computing the Nielsen number. This corresponds to just adding the dummy equation $z_0 = 0$ and adding a dominating term only to the equation with no unique dominating term. Obviously the Nielsen number obtained in this way is smaller and therefore we save computational effort for additional paths.

One can show that the additional solutions introduced by either of the above methods lead to solutions at infinity (or to solution paths diverging to infinity).

We finish this section by mentioning results for arbitrary polynomial systems of m equations in n unknowns, i.e., we consider systems of the form

$$f_j(z_1, \ldots, z_n) = 0, \quad j = 1, 2, \ldots, m.$$

We assume that each equation has a unique dominating term. We then have an $m \times n$ matrix of exponents.

For systems with $m < n$ the Nielsen number is given by $N =$ g.c.d. of determinants of all $m \times m$ submatrices. Systems with $m > n$ are ill-conditioned and the Nielsen number is zero (algebraic geometry gives results for such systems, but they are unsuitable for a homotopy approach).

13. Conclusions

Given a system of nonlinear equations

$$f_j(z_1, \ldots, z_n) = 0, \quad j = 1, 2, \ldots, m,$$

all the information about the solutions of such a system is already contained in the system. Looking at the interplay between mathematics and machines, one can then try to match mathematical theory to the properties of computers. On computers functions have to be approximated, e.g., by polynomials. This turns out to be beneficial for our theoretical considerations. It allows us to utilize Nielsen fixed point theory and to obtain *a priori* a finite number of solution classes. Piecewise linear structures, like simplices, are extremely useful from a mathematical point of view. Vertices of simplices are discrete points and therefore especially suited for representation on digital computers. The topological approach also makes the results insensitive to small perturbations (rounding errors). Pivoting and, in a computational context, new mathematical methods allow the

design of globally convergent algorithms. Methods based on Kuhn's algorithm allow us to obtain *a priori* error estimates and complexity results.

The reliability of the simplicial paths generated by pivoting is obviously a blessing for serious applications, especially safety critical applications. The systematic path following pivoting algorithms offer is far superior to the *ad hoc* decisions one has to make about the length of a step to be taken in some of the non-simplical algorithms proposed and which offer no guarantee of staying on the same path.

I would like to advance a solution concept, where the user asks the question, i.e., the user types in the system of nonlinear equations and nothing but the system of nonlinear equations (no guessing of starting values, etc.), mathematical theory provides for reliable and efficient algorithms, and the machine provides the numerical answer. Simplicial homotopy algorithms are a creative step in this direction.

Acknowledgements

Part of the material in this paper was developed in 1988 during a two months stay at Tsukuba University in Tsukuba Science City and I would like to thank Y. Yamamoto for providing a pleasant research environment. Many of the ideas expressed in this paper have been clarified and influenced by discussions with M. Kojima at the Tokyo Institute of Technology, with R. F. Brown at UCLA, with S. Smale at Berkeley, with L. T. Watson at VPI, and H. W. Kuhn at Princeton University. Special thanks to H. W. Kuhn for allowing me access to his collection of so-called "numerical counter examples".

In March 1991, as part of a symposium held at Tilburg University, various aspects of algorithms for finding zeros of polynomials were discussed. The revised version of this paper has been influenced by valuable comments from A. J. J. Talman and Zeke Wang. Furthermore, I would like to thank all researchers who commented on the first version of this paper.

References

1. Alexander, J. C. and J. A. Yorke (1978), The Homotopy Continuation Method: Numerically Implementable Topological Procedures, *Transactions of the A.M.S.* **242**, 271–284.
2. Allgower, E. L. and C. L. Keller (1971), A Search Routine for a Sperner Simplex, *Computing* **8**, 157–165.
3. Allgower, E. L. (1977), Application of a Fixed Point Algorithm to Nonlinear Boundary Value Problems Having Several Solutions, pp. 87–111, in Karamardian, S. (ed.), *Fixed Points – Algorithms and Applications*, Academic Press, New York.
4. Allgower, E. L. and K. Georg (1980), Simplical and Continuation Methods for Approximating Fixed Points, *SIAM Review* **22**, 28–85.
5. Allgower, E. L. and K. Georg (1990), *Numerical Continuation Methods*, Springer, Berlin.
6. Banach, S. (1922), Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales, *Fund. Math.* **3**, 133–181.

7. Bézout, É. (1779), Théorie générale des équations algébriques, Ph.D. Pierres, Paris.
8. Brent, R. P. (1971), Algorithms for Finding Zeros and Extrema of Functions without Calculating Derivatives, Report STAN-CS-71-198, Computer Science Department, Stanford University.
9. Brooks, R. B. S., R. F. Brown, J. Pak, and D. H. Taylor (1975), Nielsen Number of Maps of Tori, *Proc. A.M.S.* **52**, 398–400.
10. Brouwer, L. E. J. (1912), Über Abbildungen von Mannigfaltigkeiten, *Mathematische Annalen* **71**, 95–115.
11. Brouwer, L. E. J. (1952), An Intuitionist Correction of the Fixed-Point Theorem on the Sphere, *Proc. Roy. Soc. London, Ser. A* **213**, 1–2.
12. Brown, R. F. (1971), *The Lefschetz Fixed Point Theorem*, Scott, Foresman and Company, Glenview.
13. Cauchy, A. L. (1829), Exercises de mathématique, *Oeuvres (2)* **9**, 122.
14. Charnes, A., C. B. Garcia, and C. E. Lemke (1977), Constructive Proofs of Theorems Relating to: $F(x) = y$, with Applications, *Mathematical Programming* **12**, 328–343.
15. Chow, Shui-Nee, J. Mallet-Paret, and J. A. Yorke (1978), Finding Zeroes of Maps: Homotopy Methods that are Constructive with Probability One, *Mathematics of Computation* **32**, 887–899.
16. Cottle, R. W. (1982), Minimal Triangulation of the 4-Cube, *Discrete Mathematics* **40**, 25–29.
17. Dang, Ch. (1989), The D_1-Triangulation of R^n for Simplicial Algorithms for Computing Solutions on Nonlinear Equations, Tilburg University, Center for Economic Research, Discussion paper No. 8928.
18. Dang, Ch. (1990), The D2-Triangulation for Simplicial Homotopy Algorithms for Computing Solutions of Nonlinear Equations, Tilburg University, Center for Economic Research, Discussion paper No. 9024.
19. Dang, Ch. and D. Talman (1990), The D_1-Triangulation in Simplicial Variable Dimension Algorithms for Computing Solutions of Nonlinear Equations, Tilburg University, Center for Economic Research, Discussion paper No. 9027.
20. Dang, Ch. (1990), The D_3-Triangulation for Simplicial Deformation Algorithms for Computing Solutions of Nonlinear Equations, Tilburg University, Center for Economic Research, Discussion paper No. 8949.
21. Davenport, J. H., Y. Siret *et al.* (1988), *Computer Algebra*, Academic Press, London.
22. Debreu, G. (1959), *Theory of Value*, Yale University Press, New Haven.
23. Deschamps, P. J. (1977), Pricing for Congestion in Telephone Networks: A Numerical Example, pp. 473–494, in Karamardian, S. (ed.), *Fixed Points – Algorithms and Applications*, Academic Press, New York.
24. Drexler, F. J. (1978), A Homotopy Method for the Calculation of All Zeros of Zero-Dimensional Polynomial Ideals, pp. 69–93, in Wacker, H. (ed.), *Continuation Methods*, Academic Press, New York.
25. Dugundji, J. and A. Granas (1982), *Fixed Point Theory, Vol. 1*, PWN – Polish Scientific Publishers, Warszawa.
26. Eaves, B. C. (1971), Computing Kakutani Fixed Points, *SIAM J. Appl. Math.* **21**, 236–244.
27. Eaves, B. C. (1984), *A Course in Triangulations for Solving Equations with Deformations*, Springer, Berlin.
28. Eaves, B. C., F. J. Gould, H. O. Peitgen, and M. J. Todd (1983), *Homotopy Methods and Global Convergence*, Plenum Press, New York.
29. Fan, Ky (1952), A Generalization of Tucker's Combinatorial Lemma with Topological Applications, *Ann. of Math.* **56**, 431–437.
30. Fisher, M. L., F. J. Gould, and J. W. Tolle (1976), A Simplicial Approximation Algorithm for Solving Systems of Nonlinear Equations, Istituto Nazionale Di Alta Mathematica, *Symposia Mathematica* **19**, 73–90.
31. Forster, W. (ed.) (1980), *Numerical Solution of Highly Nonlinear Problems*, North-Holland, Amsterdam.
32. Forster, W. (1980), Fifty Years of Further Development of a Combinatorial Lemma, Part C, in Forster, W. (ed.), *Numerical Solution of Highly Nonlinear Problems*, pp. 215–217, North-Holland, Amsterdam.
33. Forster, W. (1984), Utilizing the Nielsen Number for Finding Multiple Solutions of Systems of Nonlinear Equations by Simplicial Fixed Point Algorithms, pp. 50–54, in Abstracts of the IIASA

Workshop on Nondifferentiable Optimization: Motivations and Applications, International Institute for Applied Systems Analysis, Laxenburg, Austria.
34. Forster, W. (1987), Computing 'All' Solutions of Systems of Polynomial Equations by Simplicial Fixed Point Algorithms, pp. 39–57, in Talman, A. J. J. and G. van der Laan (eds.), *The Computation and Modelling of Economic Equilibria*, North-Holland, Amsterdam.
35. Forster, W. (1988), Simplicial Fixed Point Algorithms and Systems of Polynomial Equations, p. 167, in Abstracts, The 13th International Symposium on Mathematical Programming, Tokyo.
36. Freund, R. M. (1986), Combinatorial Theorems on the Simplotope that Generalize Results on the Simplex and Cube, *Math. Oper. Res.* **11**, 169–179.
37. Garcia, C. B. and W. I. Zangwill (1979), Determining All Solutions to Certain Systems of Nonlinear Equations, *Mathematics of Operations Research* **4**, 1–14.
38. Garcia, C. B. and W. I. Zangwill (1979), Finding All Solutions to Polynomial Systems and Other Systems of Equations, *Mathematical Programming* **16**, 159–176.
39. Garcia, C. B. and W. I. Zangwill (1980), Global Continuation Methods for Finding All Solutions to Polynomial Systems of Equations in n Variables, pp. 481–497, in Fiacco, A. V. and K. O. Kortanek (eds.), *Extremal Methods in Systems Analysis*, Lecture Notes in Economics and Mathematical Systems No 174, Springer, Berlin.
40. Gould, F. J. (1980), Recent and Past Developments in the Simplicial Approximation Approach to Solving Nonlinear Equations – A Subjective View, pp. 466–480, in Fiacco, A. V. and K. O. Kortanek (eds.), *Extremal Methods in Systems Analysis*, Lecture Notes in Economics and Mathematical Systems No 174, Springer, Berlin.
41. Gould, F. J. and J. W. Tolle (1983), *Complementary Pivoting on a Pseudomanifold Structure with Applications in the Decision Sciences*, Heldermann Verlag, Berlin.
42. Guillemin, V. and A. Pollack (1974), *Differential Topology*, Prentice Hall, Englewood Cliffs.
43. Hartree, D. R. (1958), *Numerical Analysis*, Clarendon Press, Oxford.
44. Van der Heiden, L. (1979), Refinement Methods for Computing Fixed Points Using Primitive Sets, Yale University, Ph.D. thesis.
45. Hopf, H. (1927), Über Mindestzahlen von Fixpunkten, *Mathematische Zeitschrift* **26**, 762–774.
46. Istratescu, V. I. (1981), *Fixed Point Theory*, D. Reidel, Dordrecht.
47. Jiang, Boju (1980), *Lectures on Nielsen Fixed Point Theory*, American Mathematical Society, Providence.
48. Kakutani, S. (1941), A Generalization of Brouwer's Fixed Point Theorem, *Duke Mathematical Journal* **8**, 457–459.
49. Karamardian, S. (1977), *Fixed Points – Algorithms and Applications*, Academic Press, New York.
50. Kojima, M. (1978), A Modification of Todd's Triangulation J_3, *Math. Prog.* **15**, 223–227.
51. Kojima, M. and Y. Yamamoto (1982), Variable Dimension Algorithms: Basic Theory, Interpretation, and Extensions of Some Existing Methods, *Mathematical Programming* **24**, 177–215.
52. Kojima, M. and S. Mizuno (1983), Computation of All Solutions to a System of Polynomial Equations, *Mathematical Programming* **25**, 131–157.
53. Kojima, M. (1988), Verbal Communication During a Meeting at the Tokyo Institute of Technology.
54. Krasnosels' Kii, M. A. (1964), *Positive Solutions of Operator Equations*, P. Noordhoff, Groningen.
55. Kuhn, H. W. (1974), A New Proof of the Fundamental Theorem of Algebra, *Mathematical Programming Study* **1**, 148–158.
56. Kuhn, H. W. and J. G. MacKinnon (1975), The Sandwich Method for Finding Fixed Points, *J. Optimization Theory and Applications* **17**, 189–204.
57. Kuhn, H. W. (1977), Finding Roots of Polynomials by Pivoting, pp. 11–39, in Karamardian, S. (ed.), *Fixed Points – Algorithms and Applications*, Academic Press, New York.
58. Kuhn, H. W., Z. Wang, and S. Xu (1984), On the Cost of Computing Roots of Polynomials, *Mathematical Programming* **28**, 156–163.
59. Kuhn, H. W. (1988), Verbal Communication During a Meeting at Princeton University.
60. Van der Laan, G. (1980), *Simplicial Fixed Point Algorithms*, Mathematisch Centrum, Amsterdam.
61. Van der Laan, G. and A. J. J. Talman (1982), On the Computation of Fixed Points in the

Product Space of Unit Simplices and an Application to Noncooperative N-Person Games, *Math. Oper. Res.* **7**, 1–13.
62. Van der Laan, G., A. J. J. Talman, and L. van der Heyden (1987), Simplicial Variable Dimension Algorithms for Solving the Nonlinear Complementarity Problem on a Product of Unit Simplices Using General Labelling, *Math. Oper. Res.* **12**, 377–397.
63. Li, T. Y. and N. H. Rhee (1988), *Homotopy Algorithm for Symmetric Eigenvalue Problems*, preprint.
64. Mara, P. S. (1976), Triangulations for the Cube, *J. of Combinatorial Theory (A)* **20**, 170–177.
65. Maunder, C. R. F. (1970), *Algebraic Topology*, Van Nostrand, New York.
66. Merrill, O. H. (1972), Applications and Extensions of an Algorithm that Computes Fixed Points of Certain Upper Semi-Continuous Point to Set Mappings, Univ. of Michigan, Ph.D. thesis.
67. Miller, M. H. and J. E. Spencer (1977), The Static Economic Effects of the U.K. Joining the E.E.C.: A General Equilibrium Approach, *The Review of Economic Studies* **44**, 71–93.
68. Milnor, J. W. (1965), *Topology from the Differentiable Viewpoint*, The University Press of Virginia, Charlottesville.
69. Morgan, A. (1987), *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*, Prentice Hall, Englewood Cliffs.
70. Nielsen J. (1921), Über die Minimalzahl der Fixpunkte bei den Abbildungstypen der Ringflächen, *Mathematische Annalen* **82**, 83–93.
71. Ortega, J. M. and W. C. Rheinboldt (1970), *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York.
72. Peitgen, H. O., D. Saupe *et al.* (1984), Cayley's Problem and Julia Sets, *The Mathematical Intelligencer* **6**, 11–20.
73. Poincaré, H. (1884), Sur certaines solutions particulières du problème des trois corps, *Bulletin Astronomique* **1**, 65–74.
74. Press, W. H., B. P. Flannery *et al.* (1989), *Numerical Recipes in PASCAL*, Cambridge University Press, Cambridge.
75. Renegar, J. (1985), On the Complexity of a Piecewise Linear Algorithm for Approximating Roots of Complex Polynomials, *Mathematical Programming* **32**, 301–318
76. Renegar, J. (1985), On the Cost of Approximating all Roots of a Complex Polynomial, *Mathematical Programming* **32**, 319–336.
77. Rheinboldt, W. C. (1986), *Numerical Analysis of Parameterized Nonlinear Equations*, Wiley-Interscience, New York.
78. Rice, J. R. (1983), *Numerical Methods, Software, and Analysis, IMSL Reference Edition*, McGraw-Hill, New York.
79. Robinson, S. M. (1980), *Analysis and Computation of Fixed Points*, Academic Press, New York.
80. Rourke, C. P. and B. J. Sanderson (1972), *Introduction to Piecewise Linear Topology*, Springer, Berlin.
81. Saigal, R. (1976), On Paths Generated by Fixed Point Algorithms, *Mathematics of Oper. Res.* **1**, 359–380.
82. Sallee, J. F. (1982), A Note on Minimal Triangulations of an n-Cube, *Discrete Applied Mathematics* **4**, 211–215.
83. Scarf, H. (1967), The Approximation of Fixed Points of a Continuous Mapping, *SIAM J. Appl. Math.* **15**, 1328–1343.
84. Scarf, H. and T. Hansen (1973), *The Computation of Economic Equilibria*, Yale University Press, New Haven.
85. Shoven, J. B. (1983), The Application of Fixed Point Methods to Economics, pp. 249–261, in Eaves, B. C. *et al.* (eds.), *Homotopy Methods and Global Convergence*, Plenum Press, New York.
86. Smale, S. (1976), A Convergent Process of Price Adjustment and Global Newton Methods, *J. of Mathematical Economics* **3**, 107–120.
87. Smale, S. (1981), The Fundamental Theorem of Algebra and Complexity Theory, *Bulletin of the A.M.S.* **4**, 1–36.
88. Smale, S. (1985), On the Efficiency of Algorithms of Analysis, *Bulletin of the A.M.S.* **13**, 87–121.
89. Spanier, E. H. (1966), *Algebraic Topology*, TATA McGraw-Hill, Bombay.

90. Sperner, E. (1928), Neuer Beweis für die Invarianz der Dimensionszahl und des Gebietes, *Abh. math. Sem. Hamburg* **6**, 265–272.
91. Sperner, E. (1980), Fifty Years of Further Development of a Combinatorial Lemma, Part B, pp. 199–214, in Forster, W. (ed.), *Numerical Solutions of Highly Nonlinear Problems*, North-Holland, Amsterdam.
92. Talman, A. J. J. (1980), *Variable Dimension Fixed Point Algorithms and Triangulations*, Mathematisch Centrum, Amsterdam.
93. Todd, M. J. (1976), *The Computation of Fixed Points and Applications*, Springer, Berlin.
94. Todd, M. J. and L. Tuncel (1990), A New Triangulation for Simplicial Algorithms, Technical Report No. 946, Cornell University, Ithaca, N.Y.
95. Tucker, A. W. (1945), Some Topological Properties of Disk and Sphere, Proc. First Can. Math. Congress, Montreal, pp. 285–309.
96. Tuy, H. (1979), Pivotal Methods for Computing Equilibrium Points: Unified Approach and New Restart Algorithm, *Mathematical Programming* **16**, 210–227.
97. Watson, L. T. (1986), Globally Convergent Homotopy Methods, *SIAM Review* **28**, 529–545.
98. Wright, A. H. (1981), The Octahedral Algorithm, A New Simplicial Fixed Point Algorithm, *Mathematical Programming* **21**, 47–69.
99. Yoseloff, M. (1974), Topological Proofs of Some Combinatorial Theorems, *J. Comb. Thery* (*A*) **17**, 95–111.
100. Zeeman, E. C. (1963), Seminar on Combinatorial Topology, IHES, Mathematics Institute, Univeristy of Warwick, Coventry.