

Probability-one homotopies in computational science

Layne T. Watson

*Departments of Computer Science and Mathematics, Virginia Polytechnic Institute & State University, Blacksburg, VA
24061-0106, United States*

Abstract

Probability-one homotopy algorithms are a class of methods for solving nonlinear systems of equations that, under mild assumptions, are globally convergent for a wide range of problems in science and engineering. Convergence theory, robust numerical algorithms, and production quality mathematical software exist for general nonlinear systems of equations, and special cases such as Brouwer fixed point problems, polynomial systems, and nonlinear constrained optimization. Using a sample of challenging scientific problems as motivation, some pertinent homotopy theory and algorithms are presented. The problems considered are analog circuit simulation (for nonlinear systems), reconfigurable space trusses (for polynomial systems), and fuel-optimal orbital rendezvous (for nonlinear constrained optimization). The mathematical software packages HOMPACT90 and POLSYS_PLP are also briefly described.

Keywords: Globally convergent homotopy method; Mathematical software; Nonlinear equations; Polynomial systems

1. Introduction

Nonlinear systems of algebraic equations are ubiquitous in science and engineering, and effective algorithms to solve them become even more important as computer simulation establishes computational science as a new scientific paradigm. Nonlinear systems come in all sizes, shapes, and flavors, and a computational scientist is properly armed with a battery of algorithms. The intent of this paper is to suggest that a class of algorithms, known as *probability-one homotopy* algorithms, should be a prominent member of such a battery. For continuous (as opposed to discrete) problems, the algorithmic approaches can roughly be categorized as “local” or “global.” Examples of local methods are the classical Newton method, the secant method, quasi-Newton methods, and endless variants of these. Examples of global methods are direct search, interval arithmetic methods, and homotopy algorithms. Sometimes local methods (e.g., trust region quasi-Newton or damped Newton) are called globally convergent, but that is misleading, since the global convergence is to a local minimum of some merit function, which is not necessarily a solution to the original nonlinear system. Discrete problems, local methods, and global methods other than homotopy are not considered here.

Much of the early work on computational homotopy algorithms was motivated by Brouwer fixed point problems: given a continuous function f from a compact, convex subset of finite dimensional Euclidean space into itself, find a fixed point $x = f(x)$. The algorithms and theory are elegant and well understood for both simplicial and continuous approaches.

For nonlinear systems of equations $F(x) = 0$ not derived from Brouwer fixed point problems, the convergence theory of homotopy algorithms is well developed in terms of properties of F . Special cases,

such as when F is a polynomial system, have a deep and rich supporting theory, and special, highly sophisticated algorithms have been devised to exploit the structure of F . This case is discussed in a later section. However, except in rare instances that usually result in polynomial systems, a physical model does not directly result in a finite dimensional nonlinear system of equations $F(x) = 0$. Rather, $F(x) = 0$ results from a discretization, approximation, or iteration step of another mathematical model of the physical phenomenon. The catch is that abstract conditions on F (for a homotopy algorithm to converge) do not easily translate into meaningful or verifiable conditions on the physical model or on the discretization/approximation/iteration process. The gap here between the physical problem characteristics and properties of the subproblems $F(x) = 0$ is considerable: not many homotopy convergence theorems are stated at the level of physical modelling or the high level processes that spawn the nonlinear systems $F(x) = 0$ to be solved.

One notable exception is the solution of nonlinear two-point boundary value problems (BVPs). Conditions on the original two-point boundary value problem itself for which an approximation $F(x) = 0$ is solvable by a globally convergent homotopy algorithm have been derived. Convergence theorems directly addressing the nonlinear two-point boundary value problem exist for approximation processes based on shooting, finite differences, collocation, and finite elements. This is significant because many physical models reduce to nonlinear two-point boundary value problems, and thus convergence theory exists for a large class of problems of interest.

Section 2 gives some background material on homotopy methods, and then Sections 3–5 illustrate some aspects of the theory and algorithms via nontrivial applications. Sections 6 and 7 discuss the software packages HOMPACT90 and POLSYS_PLP, respectively.

2. Background on probability-one globally convergent homotopies

A *homotopy* is a continuous map from the interval $[0,1]$ into a function space, where the continuity is with respect to the topology of the function space. Intuitively, a homotopy $\rho(\lambda)$ continuously deforms the function $\rho(0) = g$ into the function $\rho(1) = f$ as λ goes from 0 to 1. In this case, f and g are said to be *homotopic*. Homotopy maps are fundamental tools in topology, and provide a powerful mechanism for defining equivalence classes of functions.

Homotopies provide a mathematical formalism for describing an old procedure in numerical analysis, variously known as continuation, incremental loading, and embedding. The continuation procedure for solving a nonlinear system of equations $f(x) = 0$ starts with a (generally simpler) problem $g(x) = 0$ whose solution x_0 is known. The *continuation* procedure is to track the set of zeros of

$$\rho(\lambda, x) = \lambda f(x) + (1 - \lambda)g(x) \tag{1}$$

as λ is increased monotonically from 0 to 1, starting at the known initial point $(0, x_0)$ satisfying $\rho(0, x_0) = 0$. Each step of this tracking process is done by starting at a point $(\tilde{\lambda}, \tilde{x})$ on the zero set of ρ , fixing some $\Delta\lambda > 0$, and then solving $\rho(\tilde{\lambda} + \Delta\lambda, x) = 0$ for x using a locally convergent iterative procedure, which requires an invertible Jacobian matrix $D_x\rho(\tilde{\lambda} + \Delta\lambda, x)$. The process stops at $\lambda = 1$, since $f(\bar{x}) = \rho(1, \bar{x}) = 0$ gives a zero \bar{x} of $f(x)$. Note that continuation assumes that the zeros of ρ connect the zero x_0 of g to a zero \bar{x} of f , and that the Jacobian matrix $D_x\rho(\lambda, x)$ is invertible along the zero set of ρ ; these are strong assumptions, which are frequently not satisfied in practice.

Continuation can fail because the curve γ of zeros of $\rho(\lambda, x)$ emanating from $(0, x_0)$ may (1) have turning points, (2) bifurcate, (3) fail to exist at some λ values, or (4) wander off to infinity without reaching $\lambda = 1$. Turning points and bifurcation correspond to singular $D_x\rho(\lambda, x)$. Generalizations of continuation known as *homotopy methods* attempt to deal with cases (1) and (2), and allow tracking of γ to continue through singularities. In particular, continuation monotonically increases λ , whereas homotopy methods permit λ to both increase and decrease along γ . Homotopy methods can also fail via cases (3) or (4).

The map $\rho(\lambda, x)$ connects the functions $g(x)$ and $f(x)$, hence the use of the word ‘‘homotopy.’’ In general the homotopy map $\rho(\lambda, x)$ need not be a simple convex combination of g and f as in (1), and can involve λ nonlinearly. Sometimes λ is a physical parameter in the original problem $f(x; \lambda) = 0$, where $\lambda = 1$ is the (nondimensionalized) value of interest, although ‘‘artificial parameter’’ homotopies are generally more computationally efficient than ‘‘natural parameter’’ homotopies $\rho(\lambda, x) = f(x; \lambda)$. An example of an artificial parameter homotopy map is

$$\rho(\lambda, x) = \lambda f(x; \lambda) + (1 - \lambda)(x - a), \quad (2)$$

which satisfies $\rho(0, a) = 0$. The name ‘‘artificial’’ reflects the fact that solutions to $\rho(\lambda, x) = 0$ have no physical interpretation for $\lambda < 1$. Note that $\rho(\lambda, x)$ in (2) has a unique zero $x = a$ at $\lambda = 0$, regardless of the structure of $f(x; \lambda)$.

All four shortcomings of continuation and homotopy methods have been overcome by probability-one homotopies, proposed in 1976 by Chow, Mallet-Paret, and Yorke [3]. The supporting theory, based on differential geometry, will be reformulated in less technical jargon here.

Definition 2.1. Let $U \subset \mathbf{R}^m$ and $V \subset \mathbf{R}^p$ be open sets, and let $\rho : U \times [0, 1) \times V \rightarrow \mathbf{R}^p$ be a C^2 map. ρ is said to be *transversal to zero* if the $p \times (m + 1 + p)$ Jacobian matrix $D\rho$ has full rank on $\rho^{-1}(0)$.

The C^2 requirement is technical, and part of the definition of transversality. The basis for the probability-one homotopy theory is:

Theorem 2.2 (Parametrized Sard’s Theorem) [3]. *Let $\rho : U \times [0, 1) \times V \rightarrow \mathbf{R}^p$ be a C^2 map. If ρ is transversal to zero, then for almost all $a \in U$ the map*

$$\rho_a(\lambda, x) = \rho(a, \lambda, x)$$

is also transversal to zero.

To discuss the import of this theorem, take $U = \mathbf{R}^m, V = \mathbf{R}^p$, and suppose that the C^2 map $\rho : \mathbf{R}^m \times [0, 1) \times \mathbf{R}^p \rightarrow \mathbf{R}^p$ is transversal to zero. A straightforward application of the implicit function theorem yields that for almost all $a \in \mathbf{R}^m$, the zero set of ρ_a consists of smooth, nonintersecting curves which either (1) are closed loops lying entirely in $(0, 1) \times \mathbf{R}^p$, (2) have both endpoints in $\{0\} \times \mathbf{R}^p$, (3) have both endpoints in $\{1\} \times \mathbf{R}^p$, (4) are unbounded with one endpoint in either $\{0\} \times \mathbf{R}^p$ or in $\{1\} \times \mathbf{R}^p$, or (5) have one endpoint in $\{0\} \times \mathbf{R}^p$ and the other in $\{1\} \times \mathbf{R}^p$. Furthermore, for almost all $a \in \mathbf{R}^m$, the Jacobian matrix $D\rho_a$ has full rank at *every* point in $\rho_a^{-1}(0)$. The goal is to construct a map ρ_a whose zero set has an endpoint in $\{0\} \times \mathbf{R}^p$, and which rules out (2) and (4). Then (5) obtains, and a zero curve starting at $(0, x_0)$ is *guaranteed* to reach a point $(1, \bar{x})$. All of this holds for almost all $a \in \mathbf{R}^m$, and hence with probability one [3]. Furthermore, since $a \in \mathbf{R}^m$ can be almost any point (and, indirectly,

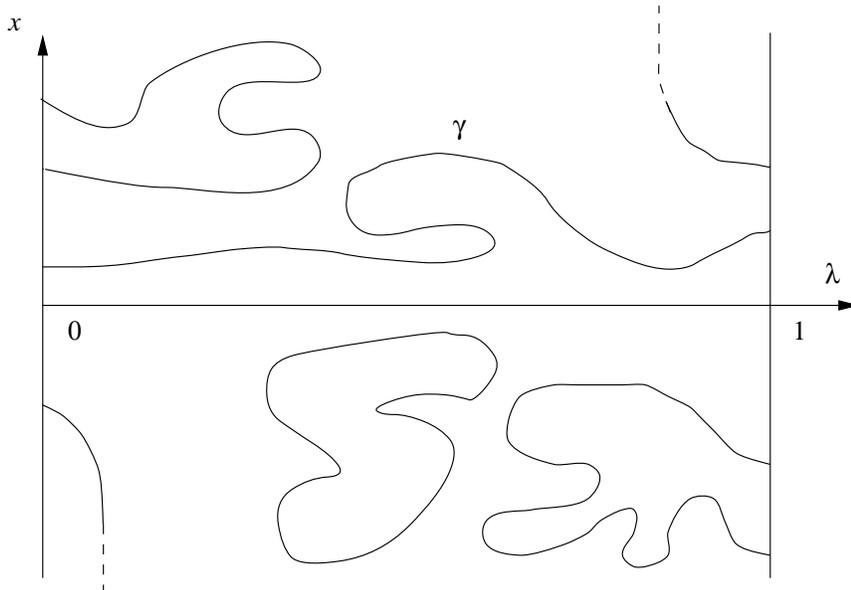


Figure 1. Zero set for $\rho_a(\lambda, x)$ satisfying properties (1)–(4).

so can the starting point x_0), an algorithm based on tracking the zero curve in (5) is legitimately called *globally convergent*. This discussion is summarized in the following theorem (and illustrated in Figure 1).

Theorem 2.3. *Let $f : \mathbf{R}^p \rightarrow \mathbf{R}^p$ be a C^2 map, $\rho : \mathbf{R}^m \times [0, 1) \times \mathbf{R}^p \rightarrow \mathbf{R}^p$ a C^2 map, and $\rho_a(\lambda, x) = \rho(a, \lambda, x)$. Suppose that*

(1) *ρ is transversal to zero,*

and, for each fixed $a \in \mathbf{R}^m$,

(2) *$\rho_a(0, x) = 0$ has a unique nonsingular solution x_0 ,*

(3) *$\rho_a(1, x) = f(x)$ ($x \in \mathbf{R}^p$).*

Then, for almost all $a \in \mathbf{R}^m$, there exists a zero curve γ of ρ_a emanating from $(0, x_0)$, along which the Jacobian matrix $D\rho_a$ has full rank. If, in addition,

(4) *$\rho_a^{-1}(0)$ is bounded,*

then γ reaches a point $(1, \bar{x})$ such that $f(\bar{x}) = 0$. Furthermore, if $Df(\bar{x})$ is invertible, then γ has finite arc length.

Any algorithm for tracking γ from $(0, x_0)$ to $(1, \bar{x})$, based on a homotopy map satisfying the hypotheses of Theorem 2.3, is called a *globally convergent probability-one homotopy algorithm*. Of course the practical numerical details of tracking γ are nontrivial, and have been the subject of twenty years of research in numerical analysis. Production quality software called HOMPACT90 [12] exists for tracking γ . The distinctions between continuation, homotopy methods, and probability-one homotopy methods are subtle but worth noting. Only the latter are provably globally convergent and (by construction) expressly avoid dealing with singularities numerically, unlike continuation and homotopy methods which must explicitly handle singularities numerically.

Assumptions (2) and (3) in Theorem 2.3 are usually achieved by the construction of ρ (such as (2)), and are straightforward to verify. Although assumption (1) is trivial to verify for some maps, if λ and a are involved nonlinearly in ρ the verification is nontrivial. Assumption (4) is typically very hard to verify, and often is a deep result, since (1)–(4) holding implies the *existence* of a solution to $f(x) = 0$.

Note that (1)–(4) are sufficient, but not necessary, for the existence of a solution to $f(x) = 0$, which is why homotopy maps not satisfying the hypotheses of Theorem 2.3 can still be very successful on practical problems. If (1)–(3) hold and a solution does *not* exist, then (4) must fail, and nonexistence is manifested by γ going off to infinity. Properties (1)–(3) are important because they guarantee good numerical properties along the zero curve γ , which, if bounded, results in a *globally convergent* algorithm. If γ is unbounded, then either the homotopy approach (with this particular ρ) has failed or $f(x) = 0$ has no solution.

A few remarks about the applicability and limitations of probability-one homotopy methods are in order. They are designed to solve a *single* nonlinear system of equations, *not* to track the solutions of a parameterized family of nonlinear systems as that parameter is varied. Thus drastic changes in the solution behavior with respect to that (natural problem) parameter have no effect on the efficacy of the homotopy algorithm, which is solving the problem for a *fixed* value of the natural parameter. In fact, it is precisely for this case of rapidly varying solutions that the probability-one homotopy approach is superior to classical continuation (which would be trying to track the rapidly varying solutions with respect to the problem parameter). Since the homotopy methods described here are not for general solution curve tracking, they are not (directly) applicable to bifurcation problems.

Homotopy methods also require the nonlinear system to be C^2 (some theory exists for piecewise C^2), and this limitation cannot be relaxed. However, requiring a finite dimensional discretization to be smooth does not mean the solution to the infinite dimensional problem must also be smooth. For example, a Galerkin formulation may produce a smooth nonlinear system in the basis function coefficients even though the basis functions themselves are discontinuous. Homotopy methods for optimization problems may converge to a local minimum or stationary point, and in this regard are no better or worse than other optimization algorithms. In special cases homotopy methods can find all the solutions if there is more than one, but in general the homotopy algorithms are only guaranteed to find one solution.

3. Analog circuit simulation

Analog circuit simulation is one application area where physical arguments directly mirror mathematical theory. Consider first a very general homotopy convergence theorem [8].

Theorem 3.1. *Let $F : \mathbf{R}^p \rightarrow \mathbf{R}^p$ be a C^2 map such that for some $r > 0$ and $\tilde{r} > 0$, $F(x)$ and $x - a$ do not point in opposite directions for $\|x\| = r$, $\|a\| < \tilde{r}$. Then F has a zero in $\{x \in \mathbf{R}^p \mid \|x\| \leq r\}$, and for almost all $a \in \mathbf{R}^p$, $\|a\| < \tilde{r}$, there is a zero curve γ of*

$$\rho_a(\lambda, x) = \lambda F(x) + (1 - \lambda)(x - a),$$

along which the Jacobian matrix $D\rho_a(\lambda, x)$ has full rank, emanating from $(0, a)$ and reaching a zero \bar{x} of F at $\lambda = 1$. Furthermore, γ has finite arc length if $DF(\bar{x})$ is nonsingular.

Note that homotopy convergence theorems are simultaneously existence theorems. The Brouwer fixed point theorem (for C^2 maps) is a special case of the next theorem [8], which is in turn a special case of Theorem 3.1.

Theorem 3.2. Let $F : \mathbf{R}^p \rightarrow \mathbf{R}^p$ be a C^2 map, and suppose there exists $r > 0$ such that $x^t F(x) \geq 0$ for $\|x\| = r$. Then F has a zero in $\{x \in \mathbf{R}^p \mid \|x\| \leq r\}$, and for almost all $a \in \mathbf{R}^p$, $\|a\| < r$, there is a zero curve γ of

$$\rho_a(\lambda, x) = \lambda F(x) + (1 - \lambda)(x - a),$$

along which the Jacobian matrix $D\rho_a(\lambda, x)$ has full rank, emanating from $(0, a)$ and reaching a zero \bar{x} of F at $\lambda = 1$. Furthermore, γ has finite arc length if $DF(\bar{x})$ is nonsingular.

A number of elegant mathematical results concern solutions to systems of equations that satisfy certain “boundedness” conditions. Perhaps the best example is the Brouwer fixed point theorem, which states that a continuous map f from a convex compact set into itself must have a fixed point; i.e., for some x^* in the set, $f(x^*) = x^*$. A fundamental problem in analog VLSI circuit simulation is to find a direct current (dc) operating point of the circuit. The voltage reference circuit [5] shown in Figure 2 is a typical example of a circuit for which standard circuit simulators have difficulty computing the dc operating points. The Brouwer fixed point theorem is applicable to the dc operating point problem. The intuition that such is the case is based on the following fact about nonlinear resistive circuits (at least, those that arise in practical integrated circuit designs). At the dc operating point of such a circuit, each node voltage is bounded in absolute value by the sum of the absolute values of the voltage sources in the circuit. A circuit with this property is called *no-gain*. In other words, if the circuit has n nodes, and the sum of the absolute values of the voltage sources is normalized to the range $[0, 1]$, then the operating point is an element of the unit n -cube. This fact is no surprise to designers of electronic circuits, although a rigorous proof of this assertion is not trivial.

The no-gain property is intrinsic to real transistors, but may or may not be preserved in a circuit simulator, depending on transistor models. An overly simple model of a transistor may not capture the *saturation behavior* of a real transistor. However, the true behavior and correct bias voltage always results from the use of a sufficiently accurate transistor model. This kind of boundedness property extends to the time domain behavior of electronic circuits. Although the output waveform may *clip*, its peak-to-peak amplitude remains bounded by the supply voltage.

Despite the historical appeal of the Brouwer fixed point theorem, and the body of knowledge about the no-gain property, Theorem 3.2 above is easier to apply to circuit equations. A physical argument for the applicability of Theorem 3.2 now follows. The nodal formulation of circuit equations specifies a sum of currents for each node. The result is a system

$$\begin{aligned} F_1(x_1, \dots, x_n) &= 0, \\ F_2(x_1, \dots, x_n) &= 0, \\ &\vdots \\ F_n(x_1, \dots, x_n) &= 0, \end{aligned}$$

where the dimension of x_i is voltage and the dimension of F_i is current. Thus, the dimension of the inner product $x^t F(x)$ is *power*.

A circuit element is *passive* if it does not generate power. This can be stated in mathematical terms by considering the voltage v_k across each element and the current i_k flowing into the element. If the sum $\sum i_k v_k$ over all elements is always nonnegative, then the device is *passive*. Passivity is a less restrictive condition than the no-gain property introduced earlier.

of 1000V. Again, common sense about electronic circuits says that an operational amplifier built using transistors and operated from +12 V and -12 V supplies cannot generate an output voltage of 1000 V. Any practical operational amplifier exhibits *limiting* behavior at its output. That is, the output is indeed equal to μv_{in} over some range of v_{in} , but beyond that range, the output voltage is bounded by the positive and negative power supply values. When a voltage amplifier is modified to model this limiting behavior (providing a more accurate model of an operational amplifier), the inner product condition of Theorem 3.2 can be satisfied at a certain radius r that may depend on the limits set for the voltage amplifiers.

It turns out that the passivity argument is not only valid for the case of nonlinear resistive circuits, i.e., circuits with no notion of time, but also applies to the time domain response of a passive circuit. Thus homotopy methods have found significant application not only to the calculation of dc operating points, but also to transient circuit analysis [5]. The description here of probability-one homotopy methods applied to analog circuit simulation is only the tip of the iceberg, since the homotopy map $\rho_a(\lambda, x) = \lambda F(x) + (1 - \lambda)(x - a)$, while it works, is not numerically efficient. Much more efficient homotopies are obtained by embedding λ deeply in the transistor models (in effect “turning on” the transistor nonlinearity as λ goes from 0 to 1), resulting in homotopy maps of the form

$$\rho_a(\lambda, x) = \lambda F(x, \lambda) + (1 - \lambda)(x - a).$$

Such homotopies are discussed in detail in [5].

4. Reconfigurable space trusses

Stewart’s platform, which has been widely adopted for use in vehicle simulators and other platform control tasks, is an example of a variable geometry truss (VGT)—a parallel-actuated manipulator. In its simplest and most elegant form, Stewart’s platform is a variable geometry octahedral truss with two triangular platforms connected by six extensible legs (see Figure 3). Such an arrangement yields an inherently strong manipulator, because all the members are loaded in pure tension or compression. In general, a variable geometry truss (VGT) can be defined as a statically determinate truss that has been modified to contain some number of variable length members. The number of these variable members is equal to the number of degrees of freedom of the device.

VGTs have been studied for their potential as adaptive or collapsing space structures. The devices proposed for these applications are typically symmetric, constructed of repeating identical cells, and have exceptional stiffness to weight ratios. Most VGTs of this type can be folded down and stored very compactly, an important feature for space applications. Some of the typical space applications that have been envisioned include booms to position equipment, berthing devices, serpentine structures to position and support a transfer tunnel, and supports for space antennae. The use of VGTs for such space and military applications has been discussed extensively in the literature.

Another interesting application of VGTs is as a manipulator arm or robot. The geometry that has been considered most suitable for this purpose is the octahedral truss. By changing the lengths of the extensible members, the manipulator arm can vary its configuration in three-dimensional space. The VGT manipulator arm can accomplish all the functions of current articulated manipulator arms, and it also has the advantage of having higher stiffness. The use of octahedral VGTs as joints in manipulators

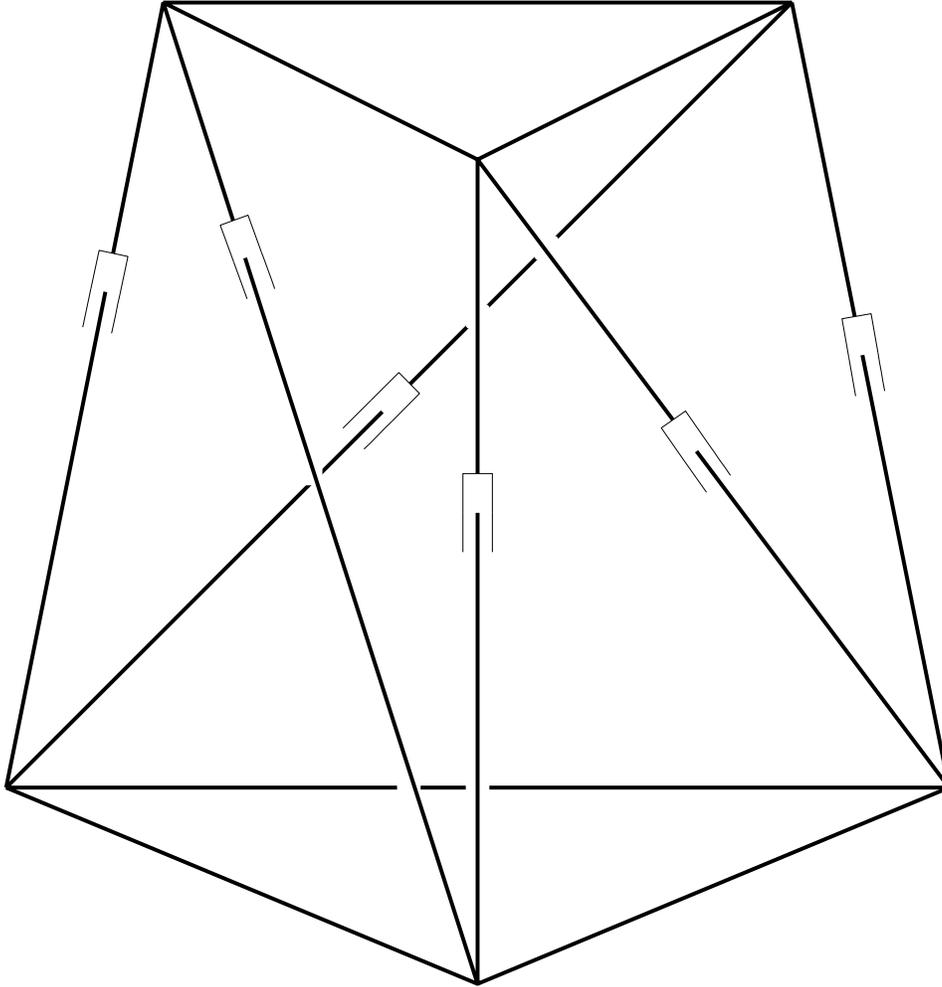


Figure 3. Stewart's platform—an example of a parallel-actuated platform manipulator.

has also been studied, and techniques, which do not require the specification of all intermediate link variables, have been developed to precisely control the end position of long chain VGT manipulators.

An ideal truss is composed exclusively of two-force members; no bending moments or torques can be transmitted at the joints. This means that the relative rotations of adjoining links must occur at a single point, either through spheric joints, or through a set of revolute joints with intersecting axes. This is a difficult requirement to satisfy exactly in practice. Many trusses are approximate in the sense that some joint offset is present and hence small bending moments are transmitted between links. All the derivations here assume that the truss under consideration is ideal, and that adjoining links are connected by spheric joints.

The octahedral VGT shown in Figure 4 is a variant of Stewart's platform used in robotics and vibration control. Although in principle it is possible to actuate any link of a VGT, this particular design has three extensible actuators in a triangle called the actuator frame. Let θ_1 , θ_2 , θ_3 be the angles made by

the links A_0A_1 , B_0B_1 , C_0C_1 , respectively, with the ground (containing the base $A_0B_0C_0$). The forward kinematics problem here is to find the angles θ_i given the 12 VGT cell link lengths. The relationship between the angles and the link lengths is

$$\begin{aligned} L_1^2 - (A \cos \theta_1 + A \cos \theta_2 + B \cos \theta_1 \cos \theta_2 - 2B \sin \theta_1 \sin \theta_2 + C) &= 0, \\ L_2^2 - (D \cos \theta_2 + D \cos \theta_3 + E \cos \theta_2 \cos \theta_3 - 2E \sin \theta_2 \sin \theta_3 + F) &= 0, \\ L_3^2 - (G \cos \theta_1 + G \cos \theta_3 + H \cos \theta_1 \cos \theta_3 - 2H \sin \theta_1 \sin \theta_3 + I) &= 0, \end{aligned} \quad (3)$$

where A, \dots, I are functions of the fixed link dimensions, and the lengths L_i are kept explicit because these links can be adjusted. Using the tangent half-angle substitutions for sine and cosine, (3) becomes

$$\begin{aligned} \alpha_1 z_1^2 + \alpha_2 z_2^2 + \alpha_3 z_1^2 z_2^2 + \alpha_4 z_1 z_2 + \alpha_5 &= 0, \\ \beta_1 z_2^2 + \beta_2 z_3^2 + \beta_3 z_2^2 z_3^2 + \beta_4 z_2 z_3 + \beta_5 &= 0, \\ \gamma_1 z_3^2 + \gamma_2 z_1^2 + \gamma_3 z_3^2 z_1^2 + \gamma_4 z_3 z_1 + \gamma_5 &= 0. \end{aligned} \quad (4)$$

Each component of (4) has degree 4, and from classical algebraic geometry the number of solutions (counting multiplicities and solutions at infinity in complex projective space, and assuming there are no solution manifolds) is the total degree $4 \cdot 4 \cdot 4 = 64$, the classical Bezout number.

Writing (4) as $F(z) = 0$, there are several important observations to be made. First, there is no easy way to separate the real solutions, complex solutions, and solutions at infinity (those existing in complex projective space but not in affine space). All known computationally feasible algorithms for solving polynomial systems like $F(z) = 0$ end up computing nonphysical complex and projective solutions. Second, a homotopy algorithm can be constructed that is guaranteed to find all 64 complex projective solutions of (4). Third, systems with structure like (4) typically have many solutions at infinity (48 in this case). Fourth, homotopy algorithms can exploit structure and avoid computing many of these solutions at infinity.

To understand this structure, partition the variables into three sets $\{z_1\}$, $\{z_2\}$, $\{z_3\}$. Observe that $F_1(z)$ has degree 2 with respect to $\{z_1\}$, treating z_2, z_3 as fixed. Similarly, $F_1(z)$ has degrees 2 and 0 with respect to $\{z_2\}$ and $\{z_3\}$, respectively. Defining d_{ij} to be the degree of $F_i(z)$ with respect to the j th variable set, the system $F(z)$ has degree structure

$$(d_{ij}) = \begin{pmatrix} 2 & 2 & 0 \\ 0 & 2 & 2 \\ 2 & 0 & 2 \end{pmatrix}.$$

Now

$$G(z) = \begin{pmatrix} (z_1^2 - 4)(z_2^2 - 9) \\ (z_2^2 - 1)(z_3^2 - 4) \\ (z_3^2 - 9)(z_1^2 - 1) \end{pmatrix} = 0 \quad (5)$$

has exactly this same degree structure, and (5) has exactly 16 finite solutions, trivially constructed. The homotopy map (ignoring the technical detail that the constants 1, 4, 9 in G must be generic)

$$\rho(\lambda, z) = \lambda F(z) + (1 - \lambda) G(z)$$

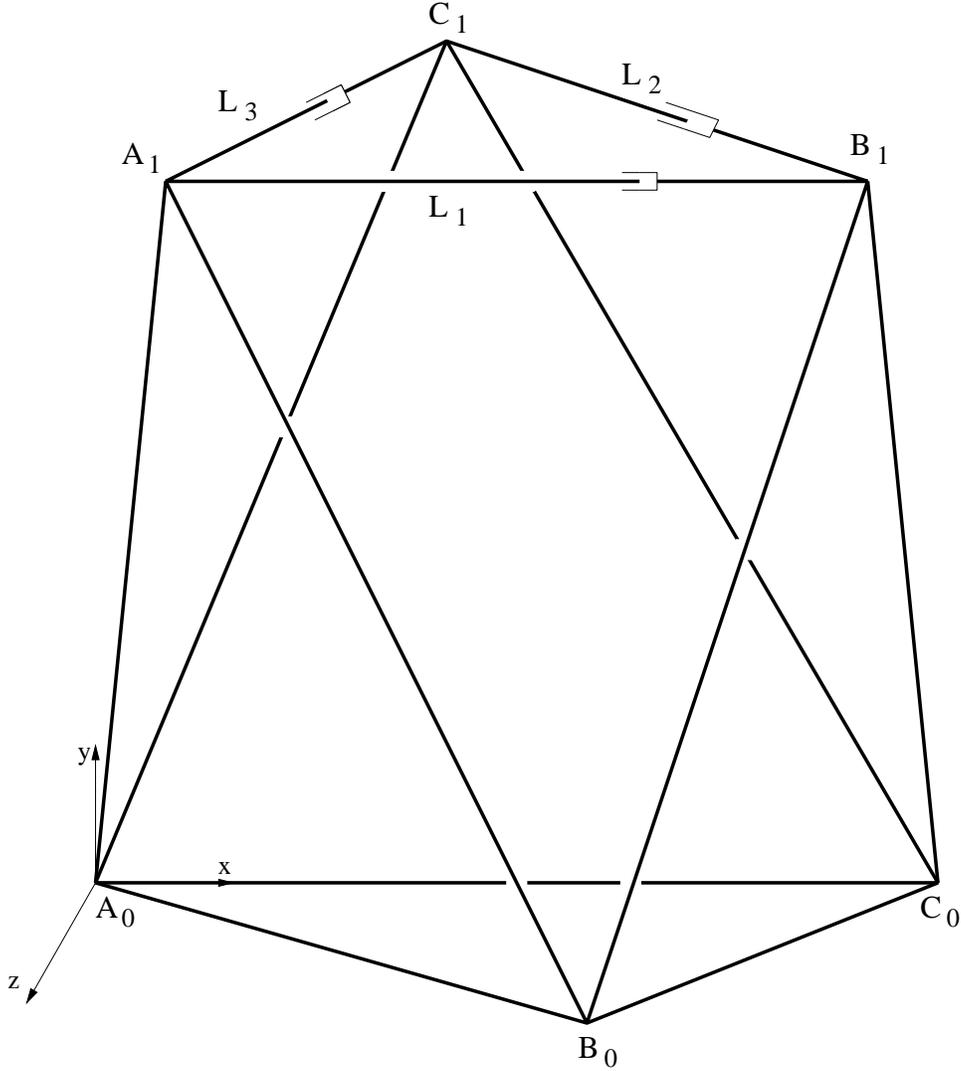


Figure 4. Octahedral variable geometry truss (VGT).

will have 16 zero curves starting at the roots of $G(z)$, and every affine solution of (4) will be reached by one of these curves. These statements follow from the general theory given below. Thus a homotopy algorithm can effectively exploit the structure in (4), and track only 16 (instead of 64) zero curves. 16 is called the *3-homogeneous Bezout number* with respect to the partition $\{\{z_1\}, \{z_2\}, \{z_3\}\}$. For typical link lengths in the octahedral VGT, there are in fact 16 different geometrical assemblies of the VGT with those links (see [2] for pictures).

The general homotopy theory for polynomial systems now follows. Let $F(z) = 0$ be a polynomial

system of n equations in n unknowns. In symbols,

$$F_i(z) = \sum_{j=1}^{n_i} \left[c_{ij} \prod_{k=1}^n z_k^{d_{ijk}} \right] = 0, \quad i = 1, \dots, n, \quad (6)$$

where the c_{ij} are complex (and usually assumed to be different from zero) and the d_{ijk} are nonnegative integers. The *degree* of $F_i(z)$ is

$$d_i = \max_{1 \leq j \leq n_i} \sum_{k=1}^n d_{ijk},$$

and the *total degree* of the system (1) is

$$d = \prod_{i=1}^n d_i.$$

Define a homotopy map $\rho : [0, 1) \times \mathbf{C}^n \rightarrow \mathbf{C}^n$ by

$$\rho(\lambda, z) = (1 - \lambda)G(z) + \lambda F(z). \quad (7)$$

$\lambda \in [0, 1)$ is the *homotopy parameter*, $G(z) = 0$ is the *start system*, and $F(z) = 0$ is the *target system*. The goal is to find a start system with the same structure as the target system, while possessing the property that $G(z) = 0$ is easily solved. Here a start system with a *partitioned linear product* (PLP) structure will be constructed.

Let $P = (P_1, P_2, \dots, P_n)$ be an n -tuple of partitions P_i of the set $\{z_1, z_2, \dots, z_n\}$. That is, for $i = 1, 2, \dots, n$, $P_i = \{S_{i1}, S_{i2}, \dots, S_{im_i}\}$, where S_{ij} has cardinality $n_{ij} \neq 0$, $\bigcup_{j=1}^{m_i} S_{ij} = \{z_1, z_2, \dots, z_n\}$, and $S_{ij_1} \cap S_{ij_2} = \emptyset$ for $j_1 \neq j_2$. For clarity, P is called the *system partition*, and the P_i are the *component partitions*. For $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m_i$ define d_{ij} to be the degree of the component F_i in only the variables of the set S_{ij} , that is, considering the variables of $\{z_1, z_2, \dots, z_n\} \setminus S_{ij}$ as constants. Thus if $F_2(z_1, z_2, z_3) = z_2^2 + z_3 z_2^3 - z_1$, $S_{21} = \{z_3\}$, and $S_{22} = \{z_1, z_2\}$, then $d_{21} = 1$, $d_{22} = 3$. It is convenient, though only for the definition of the start system, to rename the variables component-by-component. Let $S_{ij} = \{z_{ij1}, z_{ij2}, \dots, z_{ijn_{ij}}\}$. With all this said, the start system is represented mathematically by

$$G_i(z) = \prod_{j=1}^{m_i} G_{ij}, \text{ where}$$

$$G_{ij} = \begin{cases} \left(\sum_{k=1}^{n_{ij}} c_{ijk} z_{ijk} \right)^{d_{ij}} - 1, & \text{if } d_{ij} > 0; \\ 1, & \text{if } d_{ij} = 0, \end{cases} \quad i = 1, 2, \dots, n, \quad (8)$$

where the numbers $c_{ijk} \in \mathbf{C}_0 = \mathbf{C} \setminus \{0\}$ are chosen at random. The structure defined by the system partition P and manifested in (8) is called the *partitioned linear product* structure. The degree of $G_i(z)$ is

$$\deg(G_i) = \sum_{j=1}^{m_i} d_{ij}.$$

The total number of solutions to $G(z) = 0$ is called the PLP Bezout number B_{PLP} .

The significance of the system partition P , the start system $G(z) = 0$, and the PLP Bezout number B_{PLP} , is given by the following theorem from [13].

Theorem 4.1. *For almost all choices of c_{ijk} in the start system defined by (8), $\rho^{-1}(0)$ consists of B_{PLP} smooth curves emanating from $\{0\} \times \mathbf{C}^n$, which either diverge to infinity as λ approaches 1 or converge to solutions of $F(z) = 0$. Each nonsingular solution of $F(z) = 0$ will have a curve converging to it.*

These paths potentially diverging to infinity can be avoided by doing the tracking in complex projective space, with a trick due to A. P. Morgan known as the projective transformation. Define the homogenization of $F(z)$ to be

$$F'_i(w) = w_{n+1}^{d_i} F_i(w_1/w_{n+1}, \dots, w_n/w_{n+1}), \quad i = 1, \dots, n. \quad (9)$$

and that of $\rho(\lambda, z)$ to be

$$\rho'_i(\lambda, w) = w_{n+1}^{\deg(G_i)} \rho_i \left(\lambda, \frac{w_1}{w_{n+1}}, \dots, \frac{w_n}{w_{n+1}} \right), \quad i = 1, \dots, n.$$

Define the linear function

$$u(w_1, \dots, w_{n+1}) = \xi_1 w_1 + \xi_2 w_2 + \dots + \xi_{n+1} w_{n+1},$$

where the numbers $\xi_i \in \mathbf{C}_0$ are chosen at random. The *projective transformation* of $\rho(\lambda, z)$ is

$$\rho''(\lambda, w) = \begin{pmatrix} \rho'_1(\lambda, w) \\ \rho'_2(\lambda, w) \\ \vdots \\ \rho'_n(\lambda, w) \\ u(w) - 1 \end{pmatrix}.$$

$F(z)$ is defined over \mathbf{C}^n and $F'(w)$ is defined over complex projective space \mathbf{P}^n . $z \in \mathbf{C}^n$ is a solution to $F(z) = 0$ if and only if the line through $w = (z, 1)$ (a point in \mathbf{P}^n) is a solution to $F'(w) = 0$. The computer implementation actually works with ρ'' rather than with ρ . The effect of these transformations is that all the homotopy zero curves will now have finite arc length in $[0, 1] \times \mathbf{C}^{n+1}$, and every nonsingular solution of $F(z) = 0$ will be found. The precise statement follows.

Theorem 4.2. *For almost all choices of the c_{ijk} in the start system defined by (8) and almost all choices of the ξ in the linear function $u(w)$, $(\rho'')^{-1}(0)$ consists of B_{PLP} smooth curves emanating from $\{0\} \times \mathbf{C}^{n+1}$, which converge to solutions of $F'(w) = 0$. Each nonsingular solution of $F'(w) = 0$ will have a curve converging to it.*

Finally, the issue of singular solutions deserves mention. Problem symmetries result in singular solutions, and it is not uncommon in practice to have solutions with multiplicity 4, 8, or even higher. These high multiplicities result in severely rank deficient Jacobian matrices $D\rho_a(\lambda, x)$, requiring a very special “end game” to even moderately approximate these singular solutions. Very sophisticated and complicated end game strategies, based on power series or contour integrals, have been devised [13], and one of these is implemented in the code POLSYS_PLP described in Section 7.

5. Fuel-optimal orbital rendezvous

Before describing the orbital rendezvous problem and its somewhat complicated homotopy, it is useful to give some background theory for homotopies in optimization. Then, after a detailed description of the motivating application, a general homotopy convergence theory for optimization is presented.

5.1. Homotopies in optimization

A few typical convergence theorems for optimization are given here (see the survey in [7] for more examples and references). Consider first the *unconstrained optimization* problem

$$\min_x f(x). \quad (10)$$

Theorem 5.1. *Let $f : \mathbf{R}^n \rightarrow \mathbf{R}$ be a C^3 convex map with a minimum at \tilde{x} , $\|\tilde{x}\|_2 \leq M$. Then for almost all a , $\|a\|_2 < M$, there exists a zero curve γ of the homotopy map*

$$\rho_a(\lambda, x) = \lambda \nabla f(x) + (1 - \lambda)(x - a),$$

along which the Jacobian matrix $D\rho_a(\lambda, x)$ has full rank, emanating from $(0, a)$ and reaching a point $(1, \tilde{x})$, where \tilde{x} solves (10).

A function is called uniformly convex if it is convex and its Hessian's smallest eigenvalue is bounded away from zero. Consider next the constrained optimization problem

$$\min_{x \geq 0} f(x). \quad (11)$$

This is more general than it might appear because the general convex *quadratic program* reduces to a problem of the form (11).

Theorem 5.2. *Let $f : \mathbf{R}^n \rightarrow \mathbf{R}$ be a C^3 uniformly convex map. Then there exists $\delta > 0$ such that for almost all $a \geq 0$ with $\|a\|_2 < \delta$ there exists a zero curve γ of the homotopy map*

$$\rho_a(\lambda, x) = \lambda K(x) + (1 - \lambda)(x - a),$$

where

$$K_i(x) = - \left| \frac{\partial f(x)}{\partial x_i} - x_i \right|^3 + \left(\frac{\partial f(x)}{\partial x_i} \right)^3 + x_i^3,$$

along which the Jacobian matrix $D\rho_a(\lambda, x)$ has full rank, connecting $(0, a)$ to a point $(1, \bar{x})$, where \bar{x} solves the constrained optimization problem (11).

Given $F : \mathbf{R}^n \rightarrow \mathbf{R}^n$, the *nonlinear complementarity problem* is to find a vector $x \in \mathbf{R}^n$ such that

$$x \geq 0, \quad F(x) \geq 0, \quad x^t F(x) = 0. \quad (12)$$

It is interesting that homotopy methods can be adapted to deal with nonlinear inequality constraints and combinatorial conditions as in (12). Define $G : \mathbf{R}^n \rightarrow \mathbf{R}^n$ by

$$G_i(z) = -|F_i(z) - z_i|^3 + (F_i(z))^3 + z_i^3, \quad i = 1, \dots, n,$$

and let

$$\rho_a(\lambda, z) = \lambda G(z) + (1 - \lambda)(z - a).$$

Theorem 5.3. *Let $F : \mathbf{R}^n \rightarrow \mathbf{R}^n$ be a C^2 map, and let the Jacobian matrix $DG(z)$ be nonsingular at every zero of $G(z)$. Suppose there exists $r > 0$ such that $z > 0$ and $z_k = \|z\|_\infty \geq r$ imply $F_k(z) > 0$. Then for almost all $a > 0$ there exists a zero curve γ of $\rho_a(\lambda, z)$, along which the Jacobian matrix $D\rho_a(\lambda, z)$ has full rank, having finite arc length and connecting $(0, a)$ to $(1, \bar{z})$, where \bar{z} solves (12).*

Theorem 5.4. *Let $F : \mathbf{R}^n \rightarrow \mathbf{R}^n$ be a C^2 map, and let the Jacobian matrix $DG(z)$ be nonsingular at every zero of $G(z)$. Suppose there exists $r > 0$ such that $z \geq 0$ and $\|z\|_\infty \geq r$ imply $z_k F_k(z) > 0$ for some index k . Then there exists $\delta > 0$ such that for almost all $a \geq 0$ with $\|a\|_\infty < \delta$ there exists a zero curve γ of $\rho_a(\lambda, z)$, along which the Jacobian matrix $D\rho_a(\lambda, z)$ has full rank, having finite arc length and connecting $(0, a)$ to $(1, \bar{z})$, where \bar{z} solves (12).*

Homotopy algorithms for convex unconstrained optimization are generally not computationally competitive with other approaches. For constrained optimization the homotopy approach offers some advantages, and, especially for the nonlinear complementarity problem, is competitive with and often superior to other algorithms. Consider next the general nonlinear programming problem

$$\begin{aligned} & \min \theta(x) \\ & \text{subject to } g(x) \leq 0, \\ & h(x) = 0, \end{aligned} \tag{13}$$

where $x \in \mathbf{R}^n$, θ is real valued, g is an m -dimensional vector, and h is a p -dimensional vector. Assume that θ , g , and h are C^3 , and that at a local solution \bar{x} of (13), g and h satisfy some regularity condition, e.g., the weak Arrow-Hurwicz-Uzawa constraint qualification at \bar{x} . The *Kuhn-Tucker necessary optimality conditions* for (13) are

$$\begin{aligned} \nabla\theta(x) + \beta^t \nabla h(x) + \mu^t \nabla g(x) &= 0, \\ h(x) &= 0, \\ g(x) &\leq 0, \\ \mu &\geq 0, \\ \mu^t g(x) &= 0, \end{aligned} \tag{14}$$

where $\beta \in \mathbf{R}^p$ and $\mu \in \mathbf{R}^m$. The complementarity conditions $\mu \geq 0$, $g(x) \leq 0$, $\mu^t g(x) = 0$ are replaced by the equivalent nonlinear system of equations

$$W(x, \mu) = 0, \tag{15a}$$

where

$$W_i(x, \mu) = -|\mu_i + g_i(x)|^3 + \mu_i^3 - (g_i(x))^3, \quad i = 1, \dots, m. \tag{15b}$$

Thus the optimality conditions (14) take the form

$$F(x, \beta, \mu) = \begin{pmatrix} [\nabla\theta(x) + \beta^t \nabla h(x) + \mu^t \nabla g(x)]^t \\ h(x) \\ W(x, \mu) \end{pmatrix} = 0. \quad (16)$$

With $z = (x, \beta, \mu)$, the proposed homotopy map is

$$\rho_a(\lambda, z) = \lambda F(z) + (1 - \lambda)(z - a), \quad (17)$$

where $a \in \mathbf{R}^{n+p+m}$. The homotopy map in (17) has worked on some difficult realistic engineering problems, although the convergence theory for the particular map in (17) is not especially satisfying (θ , g , and h have to be very special). A more useful homotopy map is described next, and the general convergence theory presented later covers this type of map.

Frequently in practice the functions θ , g , and h involve a parameter vector c , and a solution to (13) is known for some $c = c^{(0)}$. Suppose that the problem under consideration has parameter vector $c = c^{(1)}$. Then

$$c = (1 - \lambda)c^{(0)} + \lambda c^{(1)} \quad (18)$$

parametrizes c by λ and $\theta = \theta(x; c) = \theta(x; c(\lambda))$, $g = g(x; c(\lambda))$, $h = h(x; c(\lambda))$. The optimality conditions in (16) become functions of λ as well, $F(\lambda, x, \beta, \mu) = 0$, and

$$\rho_a(\lambda, z) = \lambda F(\lambda, z) + (1 - \lambda)(z - a) \quad (19)$$

is a highly implicit nonlinear function of λ . If $F(0, z^{(0)}) = 0$, a good choice for a in practice has been found to be $a = z^{(0)}$. A natural choice for a homotopy would be simply

$$F(\lambda, z) = 0, \quad (20)$$

since the solution $z^{(0)}$ to $F(0, z) = 0$ (the problem corresponding to $c = c^{(0)}$) is known. However, for various technical reasons, (19) is much better than (20). For the orbital rendezvous problem described next, a homotopy map like (19) is used, where $c = c^{(0)}$ corresponds to a simple, relaxed constraint problem for which a solution $z^{(0)}$ is known.

5.2. An orbital mechanics problem

The problem is to find a minimum fuel rendezvous trajectory between two bodies, the nonmaneuvering target and the interceptor. The interceptor trajectory consists of Keplerian coasting arcs separated by impulsive thrusting, characterized by a change in velocity (magnitude and direction). An impulse is applied at the end of the interceptor trajectory to provide the velocity match with the target. The maneuver must be completed within some specified time and the trajectory must avoid passing through the earth, i.e., the arcs must not violate a minimum radius constraint. The fuel-optimal problem translates to minimizing the total change in the velocity (characteristic velocity).

The variables are: the coasting angles on each arc including a possible initial coast, components of the velocity change vector, and the coasting angle of the target. Assume a spherical earth and use Burdet

oscillator [2] type coordinates with the change in true anomaly as the independent variable η . Thus, the position and velocity of the body in Keplerian orbit can be represented by:

u and \hat{r} — reciprocal of the magnitude of the radius vector, and a unit vector in the radial direction;
 h and \hat{h} — magnitude of the angular momentum vector, and a unit vector along its direction;
 $\vec{r}(\eta)$ — the radius vector given by $\hat{r}(\eta)/u(\eta)$;
 $\vec{v}(\eta)$ — the velocity vector given by $h(\eta)\{u(\eta)\hat{r}'(\eta) - u'(\eta)\hat{r}(\eta)\}$.

Here $'$ refers to the derivative with respect to the change in true anomaly η . Therefore, knowing initial conditions on any subarc and the change in true anomaly, the conditions at any other point can be obtained as

$$\begin{aligned} u(\eta) &= \frac{\mu}{h^2} + \left(u(0) - \frac{\mu}{h^2}\right) \cos(\eta) + u'(0) \sin(\eta), \\ u'(\eta) &= -\left(u(0) - \frac{\mu}{h^2}\right) \sin(\eta) + u'(0) \cos(\eta), \end{aligned}$$

and similarly, the unit vectors as

$$\begin{aligned} \hat{r}(\eta) &= \hat{r}(0) \cos(\eta) + \hat{r}'(0) \sin(\eta), \\ \hat{r}'(\eta) &= -\hat{r}(0) \sin(\eta) + \hat{r}'(0) \cos(\eta). \end{aligned}$$

The time of flight T on any subarc can be obtained by integrating

$$T(\eta) = \int_0^\eta \frac{1}{hu^2(\theta)} d\theta.$$

At an impulse u and \hat{r} remain unchanged and the impulse is characterized by a change in u' , h , \hat{r}' , and \hat{h} . Thus, a change in u' and h provides the magnitude change in velocity and a change in \hat{r}' and \hat{h} provides the directional change. Since \hat{r} is fixed, the only change, if any, in \hat{r}' and \hat{h} is a rotation ϕ about \hat{r} . Using these Burdet oscillator type coordinates to represent the position and velocity, an impulse vector $\{\Delta v_x, \Delta v_y, \Delta v_z\}$ is characterized by $\{\Delta u', \Delta h, \phi\}$.

Mathematically, the aforementioned problem can be described as choosing a sequence of $\{\eta, \Delta u', \Delta h, \phi\}$ so that the characteristic velocity (total velocity change), which provides a measure of the fuel consumed, is minimized. Therefore, a time limited problem becomes: $\min_S V(x)$, where

$$S = \{(\eta, \Delta u', \Delta h, \phi)_j, \quad j = 1, \dots, nim, \quad \eta_t\},$$

nim is the prespecified number of impulses, and the characteristic velocity V can be expressed in terms of these variables as

$$V = \sum_{j=1}^{nim} \sqrt{u_{j+1}^2(0) \{h_{j+1}^2 - 2h_j h_{j+1} \cos(\phi_j) + h_j^2\} + \{\Delta h_j u'_{j+1}(0) + \Delta u'_j h_j\}^2}.$$

For the quantities u , u' , and h , the subscript j denotes the conditions at the beginning of the j th subarc, and on the variables $\Delta u'$, Δh , and ϕ the subscript j denotes the j th impulse which occurs at the end of the j th subarc. In addition, the following equality and inequality constraints must be satisfied.

Equality constraints $\mathcal{H}(x) = 0$. The conditions for rendezvous require the following position and velocity matching constraints:

- (i) final position match constraint $\mathcal{H}_1(x) \equiv \vec{r}_f - \vec{r}_t(\eta_t) = 0$;
- (ii) final velocity match constraint $\mathcal{H}_2(x) \equiv \vec{v}_f - \vec{v}_t(\eta_t) = 0$;
- (iii) time of the flight constraint $\mathcal{H}_3(x) \equiv T_f - T_t = 0$,

where the subscript f refers to the conditions on the interceptor trajectory after the final impulse and the subscript t refers to conditions on the target.

Inequality constraints $\mathcal{G}(x) \geq 0$. Additional constraints which must be satisfied along each arc of the interceptor or target trajectory in the form of an inequality are:

- (i) nonnegativity of the coasting arcs of the interceptor $\mathcal{G}_i(x) \equiv \eta_i \geq 0, i = 1, \dots, nim$;
- (ii) nonnegativity of the coasting arc of the target, $\mathcal{G}_{nim+1}(x) \equiv \eta_t \geq 0$;
- (iii) time of flight limit constraint (maximum time specified for rendezvous), $\mathcal{G}_{nim+2}(x) \equiv T_{\max} - T_f \geq 0$;
- (iv) minimum radius constraint, $\mathcal{G}_j(x) \equiv u_0 - u_{\max} \geq 0, j = nim + 3, \dots, 2nim + 1$. The transfer arc should lie outside a circle of radius $r_0 \equiv 1/u_0$. This is essentially a semi-infinite constraint, but from the nature of the transfer arcs, i.e., conic sections, the minimum radius on any subarc is given by the following:

$$\frac{1}{u_{\max}} = \text{perigee radius, if perigee passage occurs on subarc, and } \min(r_{\text{initial}}, r_{\text{final}}) \text{ otherwise.}$$

This minimum radius constraint is not C^2 . Consequently, a stiffer constraint of requiring the perigee radius of any transfer arc to be greater than the minimum allowable radius is used.

- (v) Nonnegativity of the radius constraint $\mathcal{G}_j(x) \equiv u_{\min} \geq 0, j = 2nim + 2, \dots, 3nim$. This too is a semi-infinite constraint, and the formulation here requires the final radius to be positive. This constraint is required to preclude negative distances, which are mathematically possible from the nature of the governing equations.

The orbital rendezvous problem thus has the general form of (13), namely

$$\min V(x) \quad \text{subject to} \quad -\mathcal{G}(x) \leq 0, \quad \mathcal{H}(x) = 0.$$

Homotopy convergence theory for such problems is addressed next.

5.3. Convergence theory for constrained optimization

The problem (13) involves equality constraints, and in practice (16) and (19), which include the equality constraints, have been very successful on real problems. However, a satisfactory comprehensive convergence theory does not yet exist for homotopy maps like (16) and (19). Recently, homotopy convergence theory has been developed for the inequality (only) constrained case with general θ and general inequality constraints $g(x) \leq 0$. That theory from [9] is briefly summarized here.

Let $f : \mathbf{R}^n \rightarrow \mathbf{R}$ and $g : \mathbf{R}^n \rightarrow \mathbf{R}^m$ be C^3 functions, and assume that g satisfies the Arrow-Hurwicz-Uzawa constraint qualification at every local solution of

$$\min f(x) \quad \text{subject to} \quad g(x) \leq 0. \tag{21}$$

If \bar{x} solves (21) locally, then there exists $\bar{u} \in \mathbf{R}^m$ such that (\bar{x}, \bar{u}) solves the Kuhn-Tucker problem

$$(\nabla f(x))^t + (\nabla g(x))^t u = 0, \quad (22)$$

$$g(x) \leq 0, \quad (23)$$

$$u \geq 0, \quad (24)$$

$$u^t g(x) = 0. \quad (25)$$

Let $F : \mathbf{R}^n \times [0, 1] \rightarrow \mathbf{R}$ and $G : \mathbf{R}^n \times [0, 1] \rightarrow \mathbf{R}^m$ be C^3 functions such that

$$F(x, 1) = f(x), \quad G(x, 1) = g(x), \quad (26)$$

and the optimization problem

$$\min F(x, 0) \quad \text{subject to} \quad G(x, 0) \leq 0 \quad (27)$$

has an easily obtained (local) solution x^0 . In practice $F(x, \lambda)$, $G(x, \lambda)$ represent a family of optimization problems

$$\min F(x, \lambda) \quad \text{subject to} \quad G(x, \lambda) \leq 0, \quad (28)$$

where λ is embedded deeply and nonlinearly in the objective function $F(x, \lambda)$ and constraints $G(x, \lambda)$. This embedding often embodies considerable physical insight into the problem (21), and (27) is a version of (21) with simplified physics and/or geometry. A good choice for (28) may take years to develop, and generally requires considerable problem specific knowledge and the intimate involvement of an engineer or scientist. The payoff will be a robust, globally convergent algorithm that is more efficient than applying an “off-the-shelf” algorithm, and avoids spurious solutions (e.g., unstable equilibria in mechanics or unstable circuit operating points can be expressly avoided).

One could naively solve (28) with continuation varying λ from 0 to 1, but this is precisely the point at which the probability-one theory can make a significant improvement over simple continuation in λ (and also over arc length continuation). A probability-one homotopy for (28) guarantees the existence of a zero curve γ with good numerical properties, the importance of which for practical computation cannot be overstated. A homotopy map analogous to (19) is

$$\rho(x^0, b^0, c^0, \lambda, x, u) = \left(\begin{array}{c} \lambda \left[(\nabla_x F(x, \lambda))^t + (\nabla_x G(x, \lambda))^t u \right] + (1 - \lambda)(x - x^0) \\ K(\lambda, x, u, b^0, c^0) \end{array} \right), \quad (29)$$

where

$$\begin{aligned} K_i(\lambda, x, u, b^0, c^0) = & -|(1 - \lambda)b_i^0 - G_i(x, \lambda) - u_i|^3 + ((1 - \lambda)b_i^0 - G_i(x, \lambda))^3 \\ & + u_i^3 - (1 - \lambda)c_i^0, \quad i = 1, \dots, m, \end{aligned} \quad (30)$$

is slightly different from the last m components of (19). Given arbitrary $x^0 \in \mathbf{R}^n$, choose $b^0 > 0$ such that $G(x^0, 0) - b^0 < 0$ and choose any $c^0 > 0$. The map (29), or some minor variation thereof, is what is typically used in practice, and has been extremely successful on industrial optimization problems.

The above discussion is summarized in the hypotheses of the following theorem [9]. Let $a = (x^0, b^0, c^0)$, and define $\rho_a(\lambda, x, u) = \rho(x^0, b^0, c^0, \lambda, x, u)$, according to (29) and (30). It can be proved that u^0 is always uniquely defined by $K(0, x^0, u^0, b^0, c^0) = 0$.

Theorem 5.5. *Let $f : \mathbf{R}^n \rightarrow \mathbf{R}$ and $g : \mathbf{R}^n \rightarrow \mathbf{R}^m$ be C^3 functions, let g satisfy the Arrow-Hurwicz-Uzawa constraint qualification at every local solution of (21), let $X^0 \subset \mathbf{R}^n$ and $B^0 \subset \{b \in \mathbf{R}^m \mid b > 0\}$ be open and nonempty, and for $b^0 \in B^0$ and $0 \leq \lambda \leq 1$ define*

$$S_\lambda(b^0) = \{x \in \mathbf{R}^n \mid G(x, \lambda) - (1 - \lambda)b^0 \leq 0\}.$$

For each $x^0 \in X^0$ assume there exists $b^0 \in B^0$ such that $G(x^0, 0) - b^0 < 0$. For each $x^0 \in X^0$ and $b^0 \in B^0$ satisfying $G(x^0, 0) - b^0 < 0$, further assume that $S_\lambda(b^0)$ is nonempty for $0 \leq \lambda \leq 1$, and that $\bigcup_{0 \leq \lambda \leq 1} S_\lambda(b^0)$ is bounded. Let $\rho_a(\lambda, x, u) = \rho(x^0, b^0, c^0, \lambda, x, u)$ be defined from (29) and (30). Then for

almost all $x^0 \in X^0$, almost all $b^0 \in B^0$ such that $G(x^0, 0) - b^0 < 0$, and almost all $c^0 \in \mathbf{R}^m$ with $c^0 > 0$, there exists a zero curve γ of $\rho_a(\lambda, x, u)$ emanating from $(0, x^0, u^0)$, along which the Jacobian matrix $D\rho_a(\lambda, x, u)$ has rank $n + m$. If in addition there exists $\kappa > 0$ such that for any point (λ, x, u) on γ ,

$$\|(\lambda, x, u) - (0, x^0, u^0)\| > 1 \implies \lambda \geq \kappa,$$

and for any accumulation point $(\hat{\lambda}, \hat{x})$ of (λ, x) along γ

$$\left[\nabla_x G_J(\hat{x}, \hat{\lambda}) \right] z > 0 \text{ has a solution } z,$$

where $J = \{j \mid G_j(\hat{x}, \hat{\lambda}) - (1 - \hat{\lambda})b_j^0 = 0\}$, then γ reaches a point $(1, \bar{x}, \bar{u})$, where (\bar{x}, \bar{u}) solves the Kuhn-Tucker problem (22)–(25). If rank $D\rho_a(1, \bar{x}, \bar{u}) = n + m$, then γ has finite arc length.

An interpretation of the assumptions in Theorem 5.5, and a discussion of the likelihood they might hold in practice, are given in [9]. A key component of the proof is the nature of the sets $S_\lambda(b^0)$ for $0 \leq \lambda \leq 1$, which provides considerable insight into the construction of a family of optimization problems (28) to which Theorem 5.5 applies.

6. HOMPACT90

There are several software packages implementing both continuous and *simplicial* homotopy methods; see [1] and [12] for a discussion of some of these packages. A production quality software package written in Fortran 90 is described here. *HOMPACT90* [12] is a Fortran 90 collection of codes for finding zeros or fixed points of nonlinear systems using globally convergent probability-one homotopy algorithms. Three qualitatively different algorithms—ordinary differential equation based, normal flow, quasi-Newton augmented Jacobian matrix—are provided for tracking homotopy zero curves, as well as separate routines for dense and sparse Jacobian matrices. A high level driver for the special case of polynomial systems is also provided. *HOMPACT90* features elegant interfaces, use of modules, support for several sparse matrix data structures, and modern iterative algorithms for large sparse Jacobian matrices.

HOMPACT90 is logically organized in two different ways: by algorithm/problem type and by subroutine level. There are three levels of subroutines. The top level consists of drivers, one for each problem type and algorithm type. The second subroutine level implements the major components of the algorithms such as stepping along the homotopy zero curve, computing tangents, and the end game for the solution at $\lambda = 1$. The third subroutine level handles high level numerical linear algebra such as QR

Table 1. Taxonomy of homotopy subroutines.

$x = f(x)$		$F(x) = 0$		$\rho(a, \lambda, x) = 0$		algorithm
dense	sparse	dense	sparse	dense	sparse	
FIXPDF	FIXPDS	FIXPDF	FIXPDS	FIXPDF	FIXPDS	ordinary differential equation
FIXPNF	FIXPNS	FIXPNF	FIXPNS	FIXPNF	FIXPNS	normal flow
FIXPQF	FIXPQS	FIXPQF	FIXPQS	FIXPQF	FIXPQS	augmented Jacobian matrix

factorization, and includes some LAPACK and BLAS routines. The organization of HOMPACK90 by algorithm/problem type is shown in Table 1, which lists the driver name for each algorithm and problem type.

The naming convention is

$$FIXP \left\{ \begin{matrix} D \\ N \\ Q \end{matrix} \right\} \left\{ \begin{matrix} F \\ S \end{matrix} \right\},$$

where $D \approx$ ordinary differential equation algorithm, $N \approx$ normal flow algorithm, $Q \approx$ quasi-Newton augmented Jacobian matrix algorithm, $F \approx$ dense Jacobian matrix, and $S \approx$ sparse Jacobian matrix. Depending on the problem type and the driver chosen, the user must write exactly two subroutines, whose interfaces are specified in the module HOMOTOPY, defining the problem (f or ρ). The module REAL_PRECISION specifies the real numeric model with

SELECTED_REAL_KIND(13),

which will result in 64-bit real arithmetic on a Cray, DEC VAX, and IEEE 754 Standard compliant hardware.

The special purpose polynomial system solver POLSYS1H can find all solutions in complex projective space of a *polynomial system of equations*. Since a polynomial programming problem (where the objective function, inequality constraints, and equality constraints are all in terms of polynomials) can be formulated as a polynomial system of equations, POLSYS1H can effectively find the *global optimum* of a polynomial program. However, polynomial systems can have a huge number of solutions, so this approach is only practical for small polynomial programs (e.g., surface intersection problems that arise in CAD/CAM modelling).

The organization of the Fortran 90 code into modules gives an object oriented flavor to the package. For instance, all of the drivers are encapsulated in a single MODULE HOMPACK90. The user's calling program would then simply contain a statement like

```
USE HOMPACK90, ONLY : FIXPNF
```

Many scientific programmers prefer the reverse call paradigm, whereby a subroutine returns to the calling program whenever the subroutine needs certain information (e.g., a function value) or a certain operation performed (e.g., a matrix-vector multiply). Two reverse call subroutines (STEPNX, ROOTNX) are provided for "expert" users. STEPNX is an expert reverse call stepping routine for tracking a

homotopy zero curve γ that returns to the caller for all linear algebra, all function and derivative values, and can deal gracefully with situations such as the function being undefined at the requested steplength.

ROOTNX provides an expert reverse call end game routine that finds a point on the zero curve where $g(\lambda, x) = 0$, as opposed to just the point where $\lambda = 1$. Thus ROOTNX can find turning points, bifurcation points, and other “special” points along the zero curve. The combination of STEPNX and ROOTNX provide considerable flexibility for an expert user.

7. POLSYS_PLP

The mathematical software package POLSYS_PLP [13] consists of two Fortran 90 modules (GLOBAL_PLP, POLSYS). GLOBAL_PLP contains Fortran 90 derived data types to define the target system, the start system, and the system partition. As its name suggests, GLOBAL_PLP provides data globally to the routines in POLSYS_PLP. The module POLSYS contains three subroutines: POLSYS_PLP, BEZOUT_PLP, and SINGSYS_PLP. POLSYS_PLP finds the root count (the Bezout number B_{PLP} for a given system partition P) and the roots of a polynomial system, BEZOUT_PLP finds only the root count. SINGSYS_PLP checks the singularity of a given start subsystem, and is of interest only to expert users. The package uses the HOMPACT90 modules REAL_PRECISION, HOMPACT90_GLOBAL, and HOMOTOPY [12], the HOMPACT90 subroutine STEPNX, and numerous LAPACK and BLAS subroutines. The physical organization of POLSYS_PLP into files is described in a README file that comes with the distribution.

Arguments to POLSYS_PLP include an input tracking tolerance TRACKTOL, an input final solution error tolerance FINALTOL, an input singularity tolerance SINGTOL for the root counting algorithm, input parameters for curve tracking, various output solution statistics, and four Fortran 90 optional arguments: NUMRR, RECALL, NO_SCALING, and USER_F_DF. The integer NUMRR specifies the number of iterations times 1000 that the path tracker is allowed; the default value is 1. The logical variable RECALL should be included if, after the first call, POLSYS_PLP is being called again to retrack a selected set of curves. The presence of the logical variable NO_SCALING (regardless of value) causes POLSYS_PLP *not* to scale the target polynomial system. The logical optional argument USER_F_DF specifies that the user is supplying hand-crafted code for function and Jacobian matrix evaluation—this option is recommended if efficiency is a concern, or if the original formulation of the system is other than a linear combination of monomials.

POLSYS_PLP takes full advantage of Fortran 90 features. For example, all real and complex type declarations use the KIND specification; derived data types are used for storage flexibility and simplicity; array sections, automatic arrays, and allocatable arrays are fully utilized; interface blocks are used consistently; where appropriate, modules, rather than subroutine argument lists, are used for data association; low-level linear algebra is done with Fortran 90 syntax rather than with BLAS routines; internal subroutines are used extensively with most arguments available via host association. POLSYS_PLP is easy to use, with a short argument list, and the target system $F(z)$ defined with a simple tableau format (unless the optional argument USER_F_DF is present). The calling program requires the statement

```
USE POLSYS
```

The typical use of POLSYS_PLP is either to call BEZOUT_PLP to obtain the root count B_{PLP} of a polynomial system of equations for a specified system partition P , or to call POLSYS_PLP to obtain all the roots of the polynomial (and the root count as a byproduct). It is advisable to explore several

system partitions with `BEZOUT_PLP` before committing to one and calling `POLSYS_PLP`. A sample main program `MAIN_TEMPLATE` demonstrates how to use `POLSYS_PLP` as just described. `MAIN_TEMPLATE` uses `NAMELIST` input for the target system and partition definitions, and allows the user to solve multiple polynomial systems in a single run.

The template `TARGET_SYSTEM_USER` (an external subroutine) is also provided. This subroutine would contain hand-crafted code for function and Jacobian matrix evaluation if the optional argument `USER_F_DF` to `POLSYS_PLP` were used.

The system partition must be defined by the user in the module `GLOBAL_PLP`. Heuristics exist for estimating an optimal system partition (PLP structure), but are no substitute for physical insight into the problem at hand. In practice, polynomial systems typically arise as sums of products with physical variables naturally grouped. Matching the PLP structure to the problem’s “physical” structure usually yields a near optimal Bezout number B_{PLP} . Intuitively, the idea is to get all the set degrees d_{ij} as low as possible. For real problems, an m -homogeneous [13] partition almost always suffices, and for the remainder a PLP structure is adequate. Of course linear product decomposition (LPD) [13] and general product decomposition (GPD) [13] Bezout numbers can be lower than B_{PLP} , but no class of applications has yet emerged for which B_{LPD} or B_{GPD} are significantly lower than B_{PLP} .

Acknowledgement

This work was supported in part by National Science Foundation Grant DMS-9625968, and Air Force Office of Scientific Research Grant F49620-96-1-0104.

References

- [1] Allgower, E. L., and Georg, K., *Numerical Continuation Methods*, Springer-Verlag, Berlin, 1990.
- [2] V. Arun, C. F. Reinholtz, and L. T. Watson, Application of new homotopy continuation techniques to variable geometry trusses, *J. Mech. Design* **114** (1992), 422–427.
- [3] S. N. Chow, J. Mallet-Paret, and J. A. Yorke, Finding zeros of maps: homotopy methods that are constructive with probability one, *Math. Comp.* **32** (1978), 887–899.
- [4] W. Forster, *Numerical Solution of Highly Nonlinear Problems*, North-Holland, Amsterdam, 1980.
- [5] R. C. Melville, Lj. Trajković, S.-C. Fang, and L. T. Watson, Artificial parameter homotopy methods for the DC operating point problem, *IEEE Trans. Computer-Aided Design* **12** (1993), 861–877.
- [6] G. Vasudevan, L. T. Watson, and F. H. Lutze, Homotopy approach for solving constrained optimization problems, *IEEE Trans. Automat. Control* **36** (1991), 494–498.
- [7] L. T. Watson, Globally convergent homotopy algorithms for nonlinear systems of equations, *Nonlinear Dynamics* **1** (1990), 143–191.
- [8] L. T. Watson, Numerical linear algebra aspects of globally convergent homotopy methods, *SIAM Rev.* **28** (1986), 529–545.
- [9] L. T. Watson, Theory of globally convergent probability-one homotopies for nonlinear programming, *SIAM J. Optim.*, to appear.
- [10] L. T. Watson, S. C. Billups, and A. P. Morgan, HOMPACT: a suite of codes for globally convergent homotopy algorithms, *ACM Trans. Math. Software* **13** (1987), 281–310.
- [11] L. T. Watson and R. T. Haftka, Modern homotopy methods in optimization, *Comput. Methods Appl. Mech. Engrg.* **74** (1989), 289–305.
- [12] L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, and H. F. Walker, Algorithm 777: HOMPACT90: a suite of Fortran 90 codes for globally convergent homotopy algorithms, *ACM Trans. Math. Software* **23** (1997), 514–549.
- [13] S. M. Wise, A. J. Sommese, and L. T. Watson, `POLSYS_PLP`: A partitioned linear product homotopy code for solving polynomial systems of equations, *ACM Trans. Math. Software*, to appear.