A Survey of Probability-One Homotopy Methods for Engineering Optimization

By Layne T. Watson

TR 90-47

A SURVEY OF PROBABILITY-ONE HOMOTOPY METHODS FOR ENGINEERING OPTIMIZATION

Layne T. Watson
Department of Computer Science
Virginia Polytechnic Institute & State University, Blacksburg, VA 24061 USA

Abstract. Probability-one homotopy methods are a class of algorithms for solving nonlinear systems of equations that are accurate, robust, and converge from an arbitrary starting point almost surely. These new globally convergent homotopy techniques have been successfully applied to solve Brouwer fixed point problems, polynomial systems of equations, discretizations of nonlinear two-point boundary value problems based on shooting, finite differences, collocation, and finite elements, and finite difference, collocation, and Galerkin approximations to nonlinear partial differential equations. This paper surveys the basic theory of globally convergent probability-one homotopy algorithms relevant to optimization, describes some computer algorithms and mathematical software, and applies homotopy theory to unconstrained optimization, constrained optimization, and global optimization of polynomial programs. In addition, two realistic engineering applications (optimal design of composite laminated plates and fuel-optimal orbital satellite maneuvers) are presented.

1. Introduction. Continuation in various forms has been used for a long time in mathematics and engineering, with such names as parameter continuation, incremental loading, displacement incrementation, imbedding, invariant imbedding, continuous Newton, and homotopy. The state-of-the-art of continuation methods was thoroughly surveyed in [1], and more recently in [44]. Recent mathematical developments have led to a whole new class of continuation methods known as *probability-one homotopy algorithms*, which have been successfully applied to solve Brouwer fixed point problems, polynomial systems of equations, and discretizations of nonlinear two-point boundary value problems based on shooting, finite differences, collocation, and finite elements. These new techniques have recently begun to be applied to optimization, and have found significant application in solving some engineering optimization problems.

Homotopy methods are very powerful, robust, accurate, numerically stable, and almost universally applicable, but also often prohibitively expensive. They are particularly suitable for highly nonlinear problems for which initial solution estimates are difficult to obtain. Properly implemented they are indeed *globally convergent*, i.e., converge to a solution from an *arbitrary* starting point. This (costly) global convergence feature is their forte, but also makes them inappropriate for mildly nonlinear problems or problems for which a good initial estimate of the solution is easily obtained.

The purpose of this paper is to summarize the basic theory of globally convergent homotopy methods relevant to optimization, to describe some available computer software, to show how homotopy algorithms may be applied to solve optimization problems, and to give some actual engineering applications. Section 2 gives an intuitive explanation of what is different about the new globally convergent homotopy algorithms, and Section 3 briefly recounts the basic mathematical theory. Section 4 outlines some numerical algorithms implemented in the mathematical software package HOMPACK. Sections 5 and 6 summarize basic homotopy results for optimization, and make a connection between nonlinear equations, homotopies, and optimization. Examples of the globally convergent homotopy techniques applied to optimization are given in Sections 7–11, and global optimization for polynomial programs is addressed in Section 12.

2. Continuation versus homotopy.

Continuation is a well known and established procedure in numerical analysis. The idea is to continuously deform a simple (easy) problem into the given (hard) problem, while solving the family of deformed problems. The solutions to the deformed problems are related, and can be tracked as the deformation proceeds. The function describing the deformation is called a *homotopy map*. Homotopies are a traditional part of topology, and have found significant application in nonlinear functional analysis and differential geometry. Similar ideas, such as incremental loading, are also widely used in engineering.

These traditional continuation algorithms have serious deficiencies, which have been removed by modern homotopy algorithms. The differences, however, are subtle and mathematically deep, and the mathematical proofs of the statements in this article are beyond the scope of the presentation here. To explain the differences between the old and new homotopy techniques, a more detailed discussion is required. Suppose the given problem is to find a root of the nonlinear equation $f(x) = 0$, and that $s(x) = 0$ is a simple version of the given problem with an easily obtainable unique solution x_0. Then a homotopy map could be, e.g.,

$$H(\lambda, x) = \lambda\, f(x) + (1-\lambda)\, s(x), \quad 0 \le \lambda \le 1.$$

The family of problems is $H(\lambda, x) = 0$, $0 \le \lambda \le 1$, and the idea would be to track the solutions of $H(\lambda, x) = 0$, starting from $(\lambda, x) = (0, x_0)$, as λ goes from 0 to 1. If everything worked out well, this would lead to a point $(\lambda, x) = (1, \bar{x})$, where $f(\bar{x}) = 0$. The "standard" approach is to start from a point (λ_i, x_i) with $H(\lambda_i, x_i) = 0$, and solve the problem $H(\lambda_i + \Delta\lambda, x) = 0$ for x, with $\Delta\lambda$ being a sufficiently small, fixed, positive number. The bad things that can happen are:

1) The points (λ_i, x_i) may diverge to infinity as $\lambda \to 1$.
2) The problem $H(\lambda_i + \Delta\lambda, x) = 0$ may be singular at its solution, causing numerical instability.
3) There may be no solution of $H(\lambda_i + \Delta\lambda, x) = 0$ near (λ_i, x_i).

The modern approach to homotopy methods is to construct a homotopy map $\rho_a(\lambda, x)$, involving additional parameters in the vector a, such that 1), 2), and 3) never occur or never cause any difficulty. The details of how this is done are given in the next section.

3. Homotopy theory. The theoretical foundation of all probability one globally convergent homotopy methods is given in the following differential geometry theorem:

DEFINITION. *Let E^n denote n-dimensional real Euclidean space, let $U \subset E^m$ and $V \subset E^n$ be open sets, and let $\rho : U \times [0,1) \times V \to E^n$ be a C^2 map. ρ is said to be transversal to zero if the Jacobian matrix $D\rho$ has full rank on $\rho^{-1}(0)$.*

PARAMETRIZED SARD'S THEOREM [7]. *If $\rho(a,\lambda,x)$ 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; i.e., with probability one the Jacobian matrix $D\rho_a(\lambda,x)$ has full rank on $\rho_a^{-1}(0)$.

The import of this theorem is that the zero set $\rho_a^{-1}(0)$ consists of smooth, nonintersecting curves in $[0,1) \times V$. These curves are either closed loops, or have endpoints in $\{0\} \times V$ or $\{1\} \times V$, or go to infinity. Another important consequence is that these curves have finite arc length in any compact subset of $[0,1) \times V$. The recipe for constructing a globally convergent homotopy algorithm to solve the nonlinear system of equations

$$F(x) = 0, \tag{1}$$

where $F : E^n \to E^n$ is a C^2 map, is as follows: For an open set $U \subset E^m$ construct a C^2 homotopy map $\rho : U \times [0,1) \times E^n \to E^n$ such that

1) $\rho(a,\lambda,x)$ is transversal to zero,
2) $\rho_a(0,x) = \rho(a,0,x) = 0$ is trivial to solve and has a unique solution x_0,
3) $\rho_a(1,x) = F(x)$,
4) $\rho_a^{-1}(0)$ is bounded.

Then for almost all $a \in U$ there exists a zero curve γ of ρ_a, along which the Jacobian matrix $D\rho_a$ has rank n, emanating from $(0,x_0)$ and reaching a zero $\bar{x}$ of F at $\lambda = 1$. This zero curve γ does not intersect itself, is disjoint from any other zeros of ρ_a, and has finite arc length in every compact subset of $[0,1) \times E^n$. Furthermore, if $DF(\bar{x})$ is nonsingular, then γ has finite arc length. See Figure 1.

The general idea of the algorithm is now apparent: just follow the zero curve γ emanating from $(0,x_0)$ until a zero $\bar{x}$ of $F(x)$ is reached (at $\lambda = 1$). Of course it is nontrivial to develop a viable numerical algorithm based on that idea, but at least conceptually, the algorithm for solving the nonlinear system of equations $F(x) = 0$ is clear and simple. The homotopy map (usually, but not always) is

$$\rho_a(\lambda,x) = \lambda F(x) + (1-\lambda)(x-a), \tag{2}$$

which has the same form as a standard continuation or embedding mapping. However, there are two crucial differences. First, in standard continuation, the embedding parameter λ increases

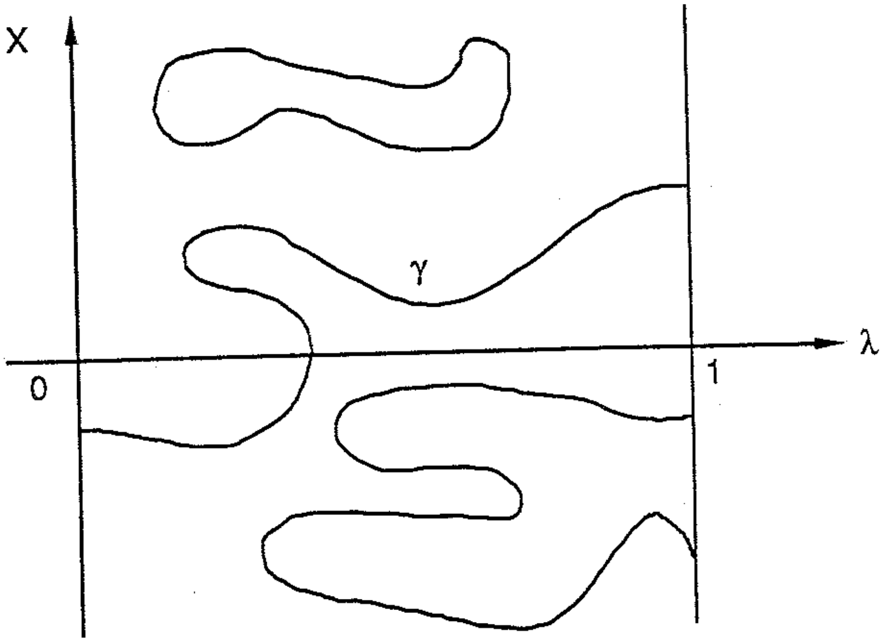

Figure 1. Zero set for $\rho_a(\lambda, x)$ satisfying properties 1) – 4).

monotonically from 0 to 1 as the trivial problem $x - a = 0$ is continuously deformed to the problem $F(x) = 0$. The present homotopy method permits λ to both increase and decrease along γ with no adverse effect; that is, turning points present no special difficulty. The second important difference is the use of the extraneous parameter a, whose consequence is that there are never any "singular points" which afflict standard continuation methods. The way in which the zero curve γ of ρ_a is followed and the full rank of $D\rho_a$ along γ guarantee this.

In order for property 4) above to hold for the homotopy map in (2), $F(x)$ and $(x - a)$ must be "asymptotically similar" (see Lemma 3 below). This is not the case for *every* $F(x)$, and so frequently other homotopy maps must be used, for example,

$$\rho_a(\lambda, x) = \lambda F(x) + (1 - \lambda) G(x; a), \tag{2a}$$

where $G(x; a)$ is a simple version of $F(x)$. For instance, $G(x; a)$ might be derived by simplifying the physical model used to derive $F(x)$. Also the homotopy map need not be a simple convex combination between $F(x)$ and $G(x; a)$; examples of homotopy maps nonlinear in λ are in [52] and [54].

The scheme just described is known as a probability-one globally convergent homotopy algorithm. The phrase "probability-one" refers to the almost any choice for a, and the "global convergence" refers to the fact that the starting point x_0 need not be anywhere near the solution $\bar{x}$. It should be emphasized that the form of the homotopy map $\rho_a(\lambda, x)$ in (2) is just a special case used here for clarity of exposition. The more general theory can be found in [39, 44, 46], and practical engineering problems requiring a ρ_a nonlinear in λ are in [52] and [54]. Below are some typical theorems for various classes of problems.

The computation of Brouwer fixed points represents one of the first successes for both simplicial [1, 34] and continuous homotopy methods [7, 39]. Brouwer fixed point problems can be very nasty, and often cause locally convergent iterative methods a great deal of difficulty.

THEOREM [39]. *Let $B = \{x \in E^n \mid \|x\|_2 = 1\}$ be the closed unit ball, and $f : B \to B$ a C^2 map. Then for almost all $a \in \operatorname{int} B$ there exists a zero curve γ of*

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

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

Typically a mathematical problem (such as a partial differential equation) results in a finite dimensional nonlinear system of equations, and what is desired are conditions on the original problem, not on the final discretized problem. Thus the results in this section are used to derive, working backwards, useful conditions on the original problem, whatever it might be. The following four lemmas, which follow from the results of [7], are used for that purpose.

LEMMA 1. *Let $g : E^p \to E^p$ be a C^2 map, $a \in E^p$, and define $\rho_a : [0, 1) \times E^p \to E^p$ by*

$$\rho_a(\lambda, y) = \lambda g(y) + (1 - \lambda)(y - a).$$

Then for almost all $a \in E^p$ there is a zero curve γ of ρ_a emanating from $(0, a)$ along which the Jacobian matrix $D\rho_a(\lambda, y)$ has full rank.

LEMMA 2. *If the zero curve γ in Lemma 1 is bounded, it has an accumulation point $(1, \bar{y})$, where $g(\bar{y}) = 0$. Furthermore, if $Dg(\bar{y})$ is nonsingular, then γ has finite arc length.*

LEMMA 3. *Let $F : E^p \to E^p$ be a C^2 map such that for some $r > 0$, $x\,F(x) \geq 0$ whenever $\|x\| = r$. Then F has a zero in $\{x \in E^p \mid \|x\| \leq r\}$, and for almost all $a \in E^p$, $\|a\| < r$, 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.

Lemma 3 is a special case of the following more general lemma.

LEMMA 4. *Let $F : E^p \to E^p$ 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 $\{x \in E^p \mid \|x\| \leq r\}$, and for almost all $a \in E^p$, $\|a\| < \tilde{r}$, 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.

These theoretical algorithms have been implemented in production quality mathematical software packages such as PITCON [33], CONKUB [20], and HOMPACK [46]. The latter, described in the next section, is an extensive collection of FORTRAN 77 rcutines implementing three different tracking algorithms for problems with both dense and sparse Jacobian matrices, and containing high level drivers for special classes of problems.

4. Curve tracking algorithms and HOMPACK. The zero curve γ of the homotopy map $\rho_a(\lambda, x)$ (of which (2) is a special case) can be tracked by many different techniques; refer to the excellent survey [1] and recent work [44], [45]. There are three primary algorithmic approaches to tracking γ that have been used in HOMPACK [46], a software package developed at Sandia National Laboratories, General Motors Research Laboratories, Virginia Polytechnic Institute and State University, and The University of Michigan: 1) an ODE-based algorithm, 2) a predictor-corrector algorithm whose corrector follows the flow normal to the Davidenko flow (a "normal flow" algorithm); 3) a version of Rheinboldt's linear predictor, quasi-Newton corrector algorithm [6], [33], (an "augmented Jacobian matrix" method).

Ordinary differential equation-based algorithm. Assuming that $F(x)$ is C^2 and a is such that ρ_a is transversal to zero, the zero curve γ is C^1 and can be parametrized by arc length s. Thus $\lambda = \lambda(s)$, $x = x(s)$ along γ, and

$$\rho_a(\lambda(s), x(s)) = 0$$

identically in s. Therefore

$$\frac{d}{ds}\rho_a(\lambda(s), x(s)) = D\rho_a(\lambda(s), x(s)) \begin{pmatrix} \frac{d\lambda}{ds} \\ \frac{dx}{ds} \end{pmatrix} = 0, \tag{3}$$

$$\left\| \left(\frac{d\lambda}{ds}, \frac{dx}{ds} \right) \right\|_2 = 1. \tag{4}$$

With the initial conditions

$$\lambda(0) = 0, \quad x(0) = x_0, \tag{5}$$

the zero curve γ is the trajectory of the initial value problem (3–5). When $\lambda(\bar{s}) = 1$, the corresponding $x(\bar{s})$ is a zero of $F(x)$. Thus all the sophisticated ODE techniques currently available can be brought to bear on the problem of tracking γ [35], [39].

Typical ODE software requires $(d\lambda/ds, dx/ds)$ explicitly, and (3), (4) only implicitly define the derivative $(d\lambda/ds, dx/ds)$. Since the dimension of the kernel of the Jacobian matrix

$$D\rho_a(\lambda(s), x(s))$$

is one (this follows from the fact that $D\rho_a$ has full rank p by the Parametrized Sard's Theorem), the derivative $(d\lambda/ds, dx/ds)$ can be calculated from any nonzero vector $z \in \ker D\rho_a$. Note that the derivative $(d\lambda/ds, dx/ds)$ is a unit tangent vector to the zero curve γ. For computational efficiency it is imperative that the number of derivative evaluations be kept small. Complete details for solving the initial value problem (3–5) and obtaining $x(\bar{s})$ are given in [48] and [39]. A discussion of the kernel computation follows.

The Jacobian matrix $D\rho_a$ is $p \times (p+1)$ with (theoretical) rank p. The crucial observation is that the last p columns of $D\rho_a$, corresponding to $D_x\rho_a$, may not have rank p, and even if they do, some other p columns may be better conditioned. The objective is to avoid choosing p "distinguished" columns, rather to treat all columns the same (not possible for sparse matrices). There are kernel finding algorithms based on Gaussian elimination and p distinguished columns [15]. Choosing and switching these p columns is tricky, and based on *ad hoc* parameters. Also, computational experience has shown that accurate tangent vectors $(d\lambda/ds, dx/ds)$ are essential, and the accuracy of Gaussian elimination may not be good enough. A conceptually elegant, as well as accurate, algorithm is to compute the QR factorization with column interchanges [44] of $D\rho_a$,

$$Q\, D\rho_a\, P^t P z = \begin{pmatrix} * & \cdots & * & * \\ & \ddots & \vdots & \vdots \\ 0 & & * & * \end{pmatrix} Pz = 0,$$

where Q is a product of Householder reflections and P is a permutation matrix, and then obtain a vector $z \in \ker D\rho_a$ by back substitution. Setting $(Pz)_{p+1} = 1$ is a convenient choice. This scheme provides high accuracy, numerical stability, and a uniform treatment of all $p+1$ columns. Finally,

$$\left(\frac{d\lambda}{ds}, \frac{dx}{ds}\right) = \pm \frac{z}{\|z\|_2},$$

where the sign is chosen to maintain an acute angle with the previous tangent vector on γ . There is a rigorous mathematical criterion, based on a $(p+1) \times (p+1)$ determinant, for choosing the sign, but there is no reason to believe that would be more robust than the angle criterion.

Several features which are a combination of common sense and computational experience should be incorporated into the algorithm. Since most ordinary differential equation solvers only control the local error, the longer the arc length of the zero curve γ gets, the farther away the computed points may be from the true curve γ. Therefore when the arc length gets too long, the last computed point $(\bar{\lambda}, \bar{x})$ is used to calculate a new parameter vector $\bar{a}$ such that

$$\rho_{\bar{a}}(\bar{\lambda}, \bar{x}) = 0 \tag{6}$$

exactly, and the zero curve of $\rho_{\bar{a}}(\lambda, x)$ is followed starting from $(\bar{\lambda}, \bar{x})$. A rigorous justification for this strategy was given in [39]. If ρ_a has the special form in (2), then trivially

$$\bar{a} = \left(\bar{\lambda}\, F(\bar{x}) + (1 - \bar{\lambda})\, \bar{x}\right) / (1 - \bar{\lambda}).$$

For more general homotopy maps ρ_a, this computation of $\bar{a}$ may be complicated.

Remember that tracking γ was merely a means to an end, namely a zero $\tilde{x}$ of $F(x)$. Since γ itself is of no interest (usually), one should not waste computational effort following it too closely.

However, since γ is the only sure way to $\bar{x}$, losing γ can be disastrous. The tradeoff between computational efficiency and reliability is very delicate, and a fool-proof strategy appears difficult to achieve. None of the three primary algorithms alone is superior overall, and each of the three beats the other two (sometimes by an order of magnitude) on particular problems. Since the algorithms' philosophies are significantly different, a hybrid will be hard to develop.

Normal flow algorithm. As the homotopy parameter vector a varies, the corresponding homotopy zero curve γ also varies. This family of zero curves is known as the Davidenko flow. The normal flow algorithm is so called because the iterates converge to the zero curve γ along the flow normal to the Davidenko flow (in an asymptotic sense).

The normal flow algorithm has four phases: prediction, correction, step size estimation, and computation of the solution at $\lambda = 1$. For the prediction phase, assume that several points $P^{(1)} = (\lambda(s_1), x(s_1))$, $P^{(2)} = (\lambda(s_2), x(s_2))$ on γ with corresponding tangent vectors $(d\lambda/ds(s_1), dx/ds(s_1))$, $(d\lambda/ds(s_2), dx/ds(s_2))$ have been found, and h is an estimate of the optimal step (in arc length) to take along γ. The prediction of the next point on γ is

$$Z^{(0)} = p(s_2 + h), \tag{7}$$

where $p(s)$ is the Hermite cubic interpolating $(\lambda(s), x(s))$ at s_1 and s_2. Precisely,

$$\begin{aligned} p(s_1) &= (\lambda(s_1), x(s_1)), \quad & p'(s_1) &= (d\lambda/ds(s_1), dx/ds(s_1)), \\ p(s_2) &= (\lambda(s_2), x(s_2)), \quad & p'(s_2) &= (d\lambda/ds(s_2), dx/ds(s_2)), \end{aligned}$$

and each component of $p(s)$ is a polynomial in s of degree less than or equal to 3.

Starting at the predicted point $Z^{(0)}$, the corrector iteration is

$$Z^{(k+1)} = Z^{(k)} - \left[D\rho_a(Z^{(k)})\right]^{\dagger} \rho_a(Z^{(k)}), \qquad k = 0, 1, \ldots \tag{8}$$

where $\left[D\rho_a(Z^{(k)})\right]^{\dagger}$ is the Moore-Penrose pseudoinverse of the $n \times (n+1)$ Jacobian matrix $D\rho_a$. Small perturbations of a produce small changes in the trajectory γ , and the family of trajectories γ for varying a is known as the "Davidenko flow". Geometrically, the iterates given by (8) return to the zero curve along the flow normal to the Davidenko flow, hence the name "normal flow algorithm".

A corrector step ΔZ is the unique minimum norm solution of the equation

$$[D\rho_a]\Delta Z = -\rho_a. \tag{9}$$

Fortunately ΔZ can be calculated at the same time as the kernel of $[D\rho_a]$, and with just a little more work. Normally for dense problems the kernel of $[D\rho_a]$ is found by computing a QR factorization of $[D\rho_a]$, and then using back substitution. By applying this QR factorization to

$-\rho_a$ and using back substitution again, a *particular* solution v to (9) can be found. Let $u \neq 0$ be any vector in the kernel of $[D\rho_a]$. Then the minimum norm solution of (9) is

$$\Delta Z = v - \frac{v^t u}{u^t u} u. \tag{10}$$

Since the kernel of $[D\rho_a]$ is needed anyway for the tangent vectors, solving (9) only requires another $O(n^2)$ operations beyond those for the kernel. The number of iterations required for convergence of (8) should be kept small (say < 4) since QR factorizations of $[D\rho_a]$ are expensive. The alternative of using $[D\rho_a(Z^{(0)})]$ for several iterations, which results in linear convergence, is rarely cost effective.

When the iteration (8) converges, the final iterate $Z^{(k+1)}$ is accepted as the next point on γ, and the tangent vector to the integral curve through $Z^{(k)}$ is used for the tangent–this saves a Jacobian matrix evaluation and factorization at $Z^{(k+1)}$. The step size estimation described next attempts to balance progress along γ with the effort expended on the iteration (8).

Define a contraction factor

$$L = \frac{\|Z^{(2)} - Z^{(1)}\|}{\|Z^{(1)} - Z^{(0)}\|}, \tag{11}$$

a residual factor

$$R = \frac{\|\rho_a(Z^{(1)})\|}{\|\rho_a(Z^{(0)})\|}, \tag{12}$$

a distance factor ($Z^* = \lim_{k\to\infty} Z^{(k)}$)

$$D = \frac{\|Z^{(1)} - Z^*\|}{\|Z^{(0)} - Z^*\|}, \tag{13}$$

and ideal values $\bar{L}$, $\bar{R}$, $\bar{D}$ for these three. Let h be the current step size (the distance from Z^* to the previous point found on γ), and $\bar{h}$ the "optimal" step size for the next step. The goal is to achieve

$$\frac{\bar{L}}{L} \approx \frac{\bar{R}}{R} \approx \frac{\bar{D}}{D} \approx \frac{\bar{h}^q}{h^q} \tag{14}$$

for some q. This leads to the choice

$$\hat{h} = \left(\min\{\bar{L}/L, \bar{R}/R, \bar{D}/D\}\right)^{1/q} h, \tag{15}$$

a worst case choice. To prevent chattering and unreasonable values, constants $h_{\min}$ (minimum allowed step size), $h_{\max}$ (maximum allowed step size), $B_{\min}$ (contraction factor), and $B_{\max}$ (expansion factor) are chosen, and $\bar{h}$ is taken as

$$\bar{h} = \min\left\{\max\{h_{\min}, B_{\min}h, \hat{h}\}, B_{\max}h, h_{\max}\right\}. \tag{16}$$

There are eight parameters in this process: $\bar{L}$, $\bar{R}$, $\bar{D}$, $h_{\min}$, $h_{\max}$, $B_{\min}$, $B_{\max}$, q. HOMPACK permits the user to specify nondefault values for any of these. The choice of $\bar{h}$ from (16) can be refined further. If (8) converged in one iteration, then $\bar{h}$ should certainly not be smaller than h, hence set

$$\bar{h} := \max\{h, \bar{h}\} \tag{17}$$

if (8) only required one iteration.

To prevent divergence from the iteration (8), if (8) has not converged after K iterations, h is halved and a new prediction is computed. Every time h is halved the old value h_{old} is saved. Thus if (8) has failed to converge in K iterations sometime during this step, the new $\bar{h}$ should not be greater than the value h_{old} known to produce failure. Hence in this case

$$\bar{h} := \min\{h_{\text{old}}, \bar{h}\}. \tag{18}$$

Finally, if (8) required the maximum K iterations, the step size should not increase, so in this case set

$$\bar{h} := \min\{h, \bar{h}\}. \tag{19}$$

The logic in (17–19) is rarely invoked, but it does have a stabilizing effect on the algorithm.

The final phase, computation of the solution at $\lambda = 1$, begins when a point $P^{(2)}$ on γ is generated such that $P_1^{(2)} \geq 1$. The solution lies somewhere on γ between the previous point $P^{(1)}$ and $P^{(2)}$. The endgame now consists of iterating until convergence the sequence of steps: inverse interpolation with the Hermite cubic (7) for $\bar{s}$ such that $p(\bar{s})_1 = 1$; two iterations of (8) starting with $Z^{(0)} = p(\bar{s})$; replacing either $P^{(1)}$ or $P^{(2)}$ by $Z^{(2)}$ such that the solution on γ is always bracketed by $P^{(1)}$ and $P^{(2)}$. A precise statement of the endgame and the convergence criterion is given in [46].

Augmented Jacobian matrix algorithm. The augmented Jacobian matrix algorithm has four major phases: prediction, correction, step size estimation, and computation of the solution at $\lambda = 1$. The algorithm here is based on Rheinboldt [33], but with some significant differences: (1) a Hermite cubic rather than a linear predictor is used; (2) a tangent vector rather than a standard basis vector is used to augment the Jacobian matrix of the homotopy map; (3) updated QR factorizations and quasi-Newton updates are used rather than Newton's method; (4) different step size control, necessitated by the use of quasi-Newton iterations, is used; (5) a different scheme for locating the target point at $\lambda = 1$ is used which allows the Jacobian matrix of F to be singular at the solution $\bar{x}$ provided rank $D\rho_a(1, \bar{x}) = n$.

The prediction phase is exactly the same as in the normal flow algorithm. Having the points $P^{(1)} = (\lambda(s_1), x(s_1)), P^{(2)} = (\lambda(s_2), x(s_2))$ on γ with corresponding tangent vectors

$$T^{(1)} = \begin{pmatrix} \dfrac{d\lambda}{ds}(s_1) \\ \dfrac{dx}{ds}(s_1) \end{pmatrix}, \qquad T^{(2)} = \begin{pmatrix} \dfrac{d\lambda}{ds}(s_2) \\ \dfrac{dx}{ds}(s_2) \end{pmatrix}, \tag{20}$$

the prediction $Z^{(0)}$ of the next point on γ is given by (7).

In order to use this predictor, a means of calculating the tangent vector $T^{(2)}$ at a point $P^{(2)}$ is required. This is done by solving the system

$$\begin{bmatrix} D\rho_a\left(P^{(2)}\right) \\ T^{(1)\,t} \end{bmatrix} z = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{21}$$

for z, where $D\rho_a$ is the $n \times (n+1)$ Jacobian of ρ_a. Normalizing z gives

$$T^{(2)} = \frac{z}{\|z\|}. \tag{22}$$

The last row of (21) insures that the tangent $T^{(2)}$ makes an acute angle with the previous tangent $T^{(1)}$. It is the augmentation of the Jacobian matrix with this additional row which motivates the name "augmented Jacobian matrix algorithm." The solution to (21) is found by computing a QR factorization of the matrix, and then using back substitution [6].

Starting with the predicted point $Z^{(0)}$, the correction is performed by a quasi-Newton iteration defined by

$$Z^{(k+1)} = Z^{(k)} - \begin{bmatrix} A^{(k)} \\ T^{(2)\,t} \end{bmatrix}^{-1} \begin{pmatrix} \rho_a\left(Z^{(k)}\right) \\ 0 \end{pmatrix}, \qquad k = 0, 1, \ldots \tag{23}$$

where $A^{(k)}$ is an approximation to the Jacobian matrix $D\rho_a\left(Z^{(k)}\right)$. The last row of the matrix in (23) insures that the iterates lie in a hyperplane perpendicular to the tangent vector $T^{(2)}$. (23) is the quasi-Newton iteration for solving the augmented nonlinear system

$$\begin{pmatrix} \rho_a(y) \\ T^{(2)\,t}\left(y - Z^{(0)}\right) \end{pmatrix} = 0. \tag{24}$$

A corrector step $\Delta Z^{(k)}$ is the unique solution to the equation

$$\begin{bmatrix} A^{(k)} \\ T^{(2)\,t} \end{bmatrix} \Delta Z^{(k)} = \begin{pmatrix} -\rho_a\left(Z^{(k)}\right) \\ 0 \end{pmatrix}. \tag{25}$$

The matrix on the left side of this equation is produced by successive Broyden rank one updates [6] of the matrix in (21). Precisely, letting $Z^{(-1)} = P^{(2)}$, $A^{(-1)} = D\rho_a\left(P^{(2)}\right)$, and

$$M^{(k)} = \begin{bmatrix} A^{(k)} \\ T^{(2)\,t} \end{bmatrix},$$

the update formulas are

$$M^{(-1)} = \begin{bmatrix} A^{(-1)} \\ T^{(2)\,t} \end{bmatrix} = \begin{bmatrix} D\rho_a\left(P^{(2)}\right) \\ T^{(1)\,t} \end{bmatrix} + e_{n+1}\left(T^{(2)} - T^{(1)}\right)^t, \tag{26}$$

and

$$M^{(k+1)} = M^{(k)} + \frac{\left(\widetilde{\Delta}\rho_a - M^{(k)}\Delta Z^{(k)}\right)\Delta Z^{(k)\,t}}{\Delta Z^{(k)\,t}\,\Delta Z^{(k)}}, \qquad k = -1, 0, \ldots \tag{27}$$

where

$$\widetilde{\Delta}\rho_a = \begin{pmatrix} \rho_a\left(Z^{(k+1)}\right) - \rho_a\left(Z^{(k)}\right) \\ 0 \end{pmatrix}.$$

These updates can be done in QR factored form, requiring a total of $\mathcal{O}(n^2)$ operations for each iteration in the correction process [6]. When the iteration (23) converges within some tolerance, the final iterate $Z^{(*)}$ is accepted as the next point on the zero curve γ.

The step size estimation algorithm is an adaptation of a procedure developed by Rheinboldt [33]. At each point $P^{(k)}$ with tangent $T^{(k)}$ along γ, the curvature is estimated by the formula

$$\left\|w^{(k)}\right\| = \frac{2}{\Delta s_k}\left|\sin\left(\alpha_k/2\right)\right|. \tag{28}$$

where

$$w^{(k)} = \frac{T^{(k)} - T^{(k-1)}}{\Delta s_k}, \qquad \alpha_k = \arccos\left(T^{(k)\,t}T^{(k-1)}\right), \qquad \Delta s_k = \left\|P^{(k)} - P^{(k-1)}\right\|.$$

Intuitively, α_k represents the angle between the last two tangent vectors, and the curvature is approximated by the Euclidean norm of the difference between these two tangents divided by Δs_k.

This curvature data can be extrapolated to produce a prediction for the curvature for the next step

$$\hat{\xi}_k = \left\|w^{(k)}\right\| + \frac{\Delta s_k}{\Delta s_k + \Delta s_{k-1}}\left(\left\|w^{(k)}\right\| - \left\|w^{(k-1)}\right\|\right). \tag{29}$$

Since $\hat{\xi}_k$ can be negative, use

$$\xi_k = \max(\xi_{min}, \hat{\xi}_k) \quad \text{for some small} \quad \xi_{min} > 0, \tag{30}$$

as the predicted curvature for the next step.

The goal in estimating the optimal step size is to keep the error in the prediction $\|Z^{(0)} - Z^{(*)}\|$ relatively constant, so that the number of iterations required by the corrector will be stable. This is achieved by choosing the step size as

$$\hat{h} = \sqrt{\frac{2\delta_k}{\xi_k}}, \tag{31}$$

where δ_k represents the ideal starting error desired for the prediction step. δ_k is chosen as a function of the tolerance for tracking the curve and is also restricted to be no larger than half of Δs_k.

As with the normal flow algorithm, additional refinements on the optimal step size are made in order to prevent chattering and unreasonable values. In particular, $\bar{h}$ is chosen to satisfy equations (16) and (18). This $\bar{h}$ is then used as the step size for the next step.

The final phase of the algorithm, computation of the solution at $\lambda = 1$, is entered when a point $P^{(2)}$ is generated such that $P_1^{(2)} \geq 1$. $P^{(2)}$ is the first such point, so the solution must lie on γ somewhere between $P^{(2)}$ and the previous point $P^{(1)}$. The algorithm for finding this solution is a two step process which is repeated until the solution is found. First, starting from a point $P^{(k)}$, a prediction $Z^{(k-2)}$ for the solution is generated such that $Z_1^{(k-2)} = 1$. Second, a single quasi-Newton iteration is performed to produce a new point $P^{(k+1)}$ close to γ, but not necessarily on the hyperplane $\lambda = 1$.

Normally, the prediction $Z^{(k-2)}$ is computed by a secant method using the last two points $P^{(k)}$ and $P^{(k-1)}$:

$$Z^{(k-2)} = P^{(k)} + \left(P^{(k-1)} - P^{(k)}\right) \frac{\left(1 - P_1^{(k)}\right)}{\left(P_1^{(k-1)} - P_1^{(k)}\right)}. \tag{32}$$

However, this formula can potentially produce a disastrous prediction (e.g., if $|P_1^{(k-1)} - P_1^{(k)}| \ll |1 - P_1^{(k)}|$), so an additional scheme is added to ensure that this does not happen. In order to implement this scheme, a point $P^{(opp)}$ must be saved. This point is chosen as the last point computed from a quasi-Newton step which is on the opposite side of the hyperplane $\lambda = 1$ from $P^{(k)}$. Thus, the points $P^{(opp)}$ and $P^{(k)}$ bracket the solution. The prediction $Z^{(k-2)}$ may be bad whenever the inequality

$$\left\| Z^{(k-2)} - P^{(k)} \right\| > \left\| P^{(k)} - P^{(opp)} \right\| \tag{33}$$

is true. In this case, $Z^{(k-2)}$ is recomputed from the equation

$$Z^{(k-2)} = P^{(k)} + \left(P^{(opp)} - P^{(k)}\right) \frac{\left(1 - P_1^{(k)}\right)}{\left(P_1^{(opp)} - P_1^{(k)}\right)}. \tag{34}$$

This chord method, while much safer than the secant method (32), is used only in the special case (33) because it has a much slower rate of convergence than the secant method.

An exception to these linear prediction schemes occurs with the first step of the final phase. Since the tangents $T^{(1)}$ and $T^{(2)}$ at $P^{(1)}$ and $P^{(2)}$ are available, this information is used to generate a Hermite cubic polynomial $p(s)$ for calculating the first prediction point $Z^{(0)}$. This is done by finding the root $\bar{s}$ of the equation $p_1(s) = 1$. $Z^{(0)}$ is then given by

$$Z^{(0)} = p(\bar{s}). \tag{35}$$

After the predictor $Z^{(k-2)}$ has been determined, a quasi-Newton step is taken to get the point $P^{(k+1)}$. This step is defined by

$$P^{(k+1)} = Z^{(k-2)} + \Delta Z^{(k-2)}, \tag{36}$$

where $\Delta Z^{(k-2)}$ is the solution to (25). Again, the matrix in (25) is produced by the rank one updates (26) and (27).

The alternating process of computing a prediction and taking a quasi-Newton step is repeated until the solution is found.

5. Basic optimization homotopies. Consider first the unconstrained optimization problem

$$\min_x f(x). \tag{37}$$

THEOREM [41]. *Let $f : E^n \to E$ be a C^3 convex map with a minimum at $\tilde{x}$, $\|\tilde{x}\|_2 \le M$. 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 (37).

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

$$\min_{x \ge 0} f(x). \tag{38}$$

This is more general than it might appear because the general convex quadratic program reduces to a problem of the form (38).

THEOREM [41]. *Let $f : E^n \to E$ be a C^3 uniformly convex map. Then there exists $\delta > 0$ such that for almost all $a \ge 0$ 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 (38).

Given $F : E^n \to E^n$, the nonlinear complementarity problem is to find a vector $x \in E^n$ such that

$$x \ge 0, \quad F(x) \ge 0, \quad x^t F(x) = 0. \tag{39}$$

At a solution $\bar{x}$, $\bar{x}$ and $F(\bar{x})$ are "complementary" in the sense that if $\bar{x}_i > 0$, then $F_i(\bar{x}) = 0$, and if $F_i(\bar{x}) > 0$, then $\bar{x}_i = 0$. This problem is difficult because there are linear constraints $x \ge 0$, nonlinear constraints $F(x) \ge 0$, and a combinatorial aspect from the complementarity condition $x^t F(x) = 0$. It is interesting that homotopy methods can be adapted to deal with nonlinear constraints and combinatorial conditions.

Define $G : E^n \to E^n$ by

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

and let

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

THEOREM [40]. *Let $F : E^n \to E^n$ 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 (39).*

THEOREM [40]. *Let $F : E^n \to E^n$ 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 (39).*

Homotopy algorithms for convex unconstrained optimization are only of theoretical interest, and are generally not computationally competitive with other approaches, but it is reassuring that the globally convergent homotopy techniques can theoretically be directly applied. For constrained optimization the homotopy approach offers some advantages, and, especially for the nonlinear complementarity problem, is competitive with other algorithms. See [47] for an application of homotopy techniques to the linear complementarity problem. Constrained optimization is addressed in the next few sections.

6. Expanded Lagrangian Homotopy. The expanded Lagrangian homotopy method of Poore [31, 32] is applicable to the general nonlinear programming problem

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

where $x \in E^n$, 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^2. In this general situation the complete formulation and solution algorithm for the expanded Lagrangian homotopy are rather complicated. The essence of the method is presented here, referring the reader to [31] and [32] for a discussion of the theoretical and practical subtleties. The technique has been applied to linear programming [31] and the linear complementarity problem [47], but is currently primarily of theoretical interest.

The expanded Lagrangian approach may be described as an optimization/continuation approach and has in its simplest form two main steps.

Step 1. (Optimization phase).

At $r = r_0 > 0$ solve the unconstrained minimization problem

$$\min_x P(x, r)$$

where

$$P(x, r) = \theta(x) + \frac{1}{2r} h(x)^t h(x) - r \sum_{i=1}^m \ln\bigl(-g_i(x)\bigr).$$

Step 2A. (Switch to expanded system).

A (local) solution of min P must satisfy

$$0 = \nabla_x P = \nabla\theta(x) + \frac{h(x)^t \nabla h(x)}{r} - \sum_{i=1}^m \frac{r}{g_i(x)} \nabla g_i(x).$$

Introduce the following variables:

$$\begin{aligned} \beta &= \frac{h(x)}{r}, \\ \mu_i &= \frac{r}{-g_i(x)}, \qquad i = 1, \ldots, m, \end{aligned}$$

which ultimately represent the Lagrange multipliers. This helps to remove the inevitable ill-conditioning associated with penalty methods for small r and we thus obtain our equivalent but expanded system:

$$\begin{aligned} \nabla\theta(x) + \beta^t \nabla h(x) + \mu^t \nabla g(x) &= 0, \\ h(x) - r\beta &= 0, \\ \mu_i g_i(x) + r &= 0, \qquad i = 1, \ldots, m. \end{aligned}$$

(Remark. As a result of the optimization phase and the initial starting point with $r_0 > 0$, the solution $x^{(0)}$ of min $P(x, r_0)$ satisfies $g(x^{(0)}) < 0$. As a consequence, $\mu^{(0)} > 0$ from the definition of μ. μ remains positive until $r = 0$ where we formally have

$$\begin{aligned} \nabla\theta(x) + \beta^t \nabla h(x) + \mu^t \nabla g(x) &= 0, \\ h(x) &= 0, \\ g(x) &\le 0, \\ \mu &\ge 0, \\ \mu_i g_i(x) &= 0, \qquad i = 1, \ldots, m, \end{aligned}$$

which implies that we have solved the problem.)

In practice we do not solve the optimization problem min P to high accuracy since a highly accurate solution may have only a digit or two in common with the final answer. However, it is imperative that ∇P be reasonably small in magnitude, say less than $r_0/10$. The expanded system is converted to a homotopy map by letting $r = r_0(1-\lambda)$ and modifying the first equation to obtain:

$$\begin{aligned} \nabla\theta(x) + \beta^t\nabla h(x) + \mu^t\nabla g(x) - \frac{r}{r_0}\nabla P(x^{(0)}, r_0) &= 0, \\ h(x) - r\beta &= 0, \\ \mu_i g_i(x) + r &= 0, \qquad i = 1, \ldots, m. \end{aligned} \tag{40}$$

Write this system of $n + p + m$ equations in the $n + p + m + 1$ variables λ, x, β, μ as

$$\Upsilon(\lambda, x, \beta, \mu) = 0.$$

Step 2B. (Track the zero curve of Υ from $r = r_0$ to $r = 0$.)

Starting with arbitrary $r_0 > 0$ and feasible interior point $x^{(0)}$ ($g(x^{(0)}) < 0$), the rest of the initial point $(0, x^{(0)}, \beta^{(0)}, \mu^{(0)})$ is given by

$$\begin{aligned} \beta^{(0)} &= \frac{h(x^{(0)})}{r_0}, \\ \mu_i^{(0)} &= \frac{r_0}{-g_i(x^{(0)})}, \qquad i = 1, \ldots, m. \end{aligned}$$

This approach requires careful attention to implementation details. For example, the linear algebra and globalization techniques with dynamic scaling are critically important in the optimization phase. For degenerate problems the path can still be long. One possible resolution is the use of shifts and weights as developed in the method of multipliers [5], but holding $r = r_0$ fixed. (This approach is currently under investigation in the context of linear programming [31].) Note that the optimization phase (Step 1) can be omitted altogether, starting Step 2B with an arbitrary interior feasible point $x^{(0)}$ ($g(x^{(0)}) < 0$), so that (40) is a true global homotopy. As a practical matter, however, it is advantageous to get a good starting point by doing Step 1 with a small r_0.

7. Application of expanded Lagrangian homotopy to the linear complementarity problem. As an illustration, the expanded Lagrangian homotopy method will be applied to the linear complementarity problem:

$$\begin{gathered} w - Mz = q, \\ w \geq 0, \quad z \geq 0, \quad w^t z = 0, \end{gathered}$$

where M is a given real $n \times n$ matrix and $q \in E^n$ is given; the unknowns are $w \in E^n$ and $z \in E^n$.

Step 1. (Optimization phase).

At $r = r_0 > 0$ solve the unconstrained minimization problem

$$\min_{w,z} P(w, z, r)$$

where

$$P(w,z,r) = \frac{1}{2r}\|w - Mz - q\|_2^2 + \frac{1}{2r}\langle w, z\rangle^2 - r\sum_{i=1}^{n}\ln z_i - r\sum_{i=1}^{n}\ln w_i.$$

Step 2A. (Switch to expanded system).

A (local) solution of $\min P$ must satisfy

$$0 = \nabla_{(w,z)}P = \begin{pmatrix} I \\ -M^t \end{pmatrix}\frac{(w - Mz - q)}{r} + \begin{pmatrix} z \\ w \end{pmatrix}\frac{\langle w, z\rangle}{r} - r\left(\frac{1}{w_1},\ldots,\frac{1}{w_n},\frac{1}{z_1},\ldots,\frac{1}{z_n}\right)^t.$$

Introduce the following variables:

$$\begin{aligned} \beta &= \frac{w - Mz - q}{r}, \\ \theta &= \frac{\langle w, z\rangle}{r}, \\ \mu_i &= \frac{r}{w_i}, \qquad i = 1,\ldots,n, \\ \eta_i &= \frac{r}{z_i}, \qquad i = 1,\ldots,n, \end{aligned}$$

which ultimately represent the Lagrange multipliers. This helps to remove the inevitable ill-conditioning associated with penalty methods for small r and we thus obtain our equivalent but expanded system:

$$\begin{aligned} \begin{pmatrix} I \\ -M^t \end{pmatrix}\beta + \begin{pmatrix} z \\ w \end{pmatrix}\theta - \begin{pmatrix} \mu \\ \eta \end{pmatrix} &= 0, \\ w - Mz - q - r\beta &= 0, \\ \langle w, z\rangle - r\theta &= 0, \\ \mu_i w_i - r &= 0, \qquad i = 1,\ldots,n, \\ \eta_i z_i - r &= 0, \qquad i = 1,\ldots,n. \end{aligned}$$

(Remark. As a result of the optimization phase and the initial starting point with $r_0 > 0$, the solution $(w^{(0)}, z^{(0)})$ of $\min P(w,z,r_0)$ satisfies $z^{(0)} > 0$ and $w^{(0)} > 0$. As a consequence, $\mu^{(0)} > 0$ and $\eta^{(0)} > 0$ from the definitions of μ and η. They remain positive until $r = 0$ where we formally have

$$\begin{aligned} \begin{pmatrix} I \\ -M^t \end{pmatrix}\beta + \begin{pmatrix} z \\ w \end{pmatrix}\theta - \begin{pmatrix} \mu \\ \eta \end{pmatrix} &= 0, \\ w - Mz - q &= 0, \\ \langle w, z\rangle &= 0, \\ \mu_i w_i &= 0, \qquad i = 1,\ldots,n, \\ \eta_i z_i &= 0, \qquad i = 1,\ldots,n, \\ w, z, \theta, \mu, \eta &\geqq 0, \end{aligned}$$

which implies that we have solved the problem.)

The expanded system is converted to a homotopy map by letting $r = r_0(1-\lambda)$ and modifying the first equation to obtain:

$$\begin{aligned}
\begin{pmatrix} I \\ -M^t \end{pmatrix}\beta + \begin{pmatrix} z \\ w \end{pmatrix}\theta - \begin{pmatrix} \mu \\ \eta \end{pmatrix} - \frac{r}{r_0}\nabla P(w^{(0)}, z^{(0)}, r_0) &= 0, \\
w - Mz - q - r\beta &= 0, \\
\langle w, z\rangle - r\theta &= 0, \\
\mu_i w_i - r &= 0, \qquad i = 1, \ldots, n, \\
\eta_i z_i - r &= 0, \qquad i = 1, \ldots, n.
\end{aligned}$$

Write this system of $5n+1$ equations in the $5n+2$ variables λ, w, z, β, θ, μ, η as

$$\Upsilon(\lambda, w, z, \beta, \theta, \mu, \eta) = 0.$$

Step 2B. (Track the zero curve of Υ from $r = r_0$ to $r = 0$.)

Starting with arbitrary $r_0 > 0$, $w^{(0)} > 0$ and $z^{(0)} > 0$, the rest of the initial point $(0, w^{(0)}, z^{(0)}, \beta^{(0)}, \theta_0, \mu^{(0)}, \eta^{(0)})$ is given by

$$\begin{aligned}
\beta^{(0)} &= \frac{w^{(0)} - Mz^{(0)} - q}{r_0}, \\
\theta_0 &= \frac{\langle w^{(0)}, z^{(0)}\rangle}{r_0}, \\
\mu_i^{(0)} &= \frac{r_0}{w_i^{(0)}}, \qquad i = 1, \ldots, n, \\
\eta_i^{(0)} &= \frac{r_0}{z_i^{(0)}}, \qquad i = 1, \ldots, n.
\end{aligned}$$

Computational experience with this approach to the LCP is reported in [47].

8. Application to optimal structural design. Composite materials are ideal for structural applications where high strength-to-weight and stiffness-to-weight ratios are required. Design optimization of composite structures has gained importance in recent years as the engineering applications of fiber-reinforced materials have increased and weight savings has become an essential design objective, especially for aircraft and spacecraft structures. The laminates considered here are symmetric about the middle surface with $2n$ layers (see Figure 2), so that the bending response is not coupled to the membrane action. The optimization problem is to maximize the buckling load of a $2n$-layered composite plate (Figure 3) for a given total plate thickness. The thickness of each layer is assumed to be constant over the plate, and for a given stacking sequence of the ply orientations, each thickness is taken as a design variable.

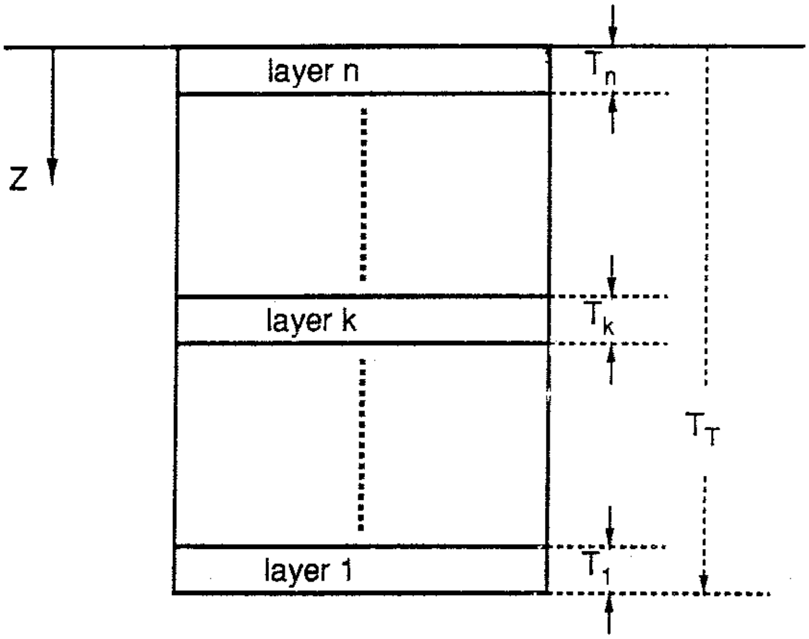

Figure 2. Geometry of half of a $2n$-layered symmetric laminate.

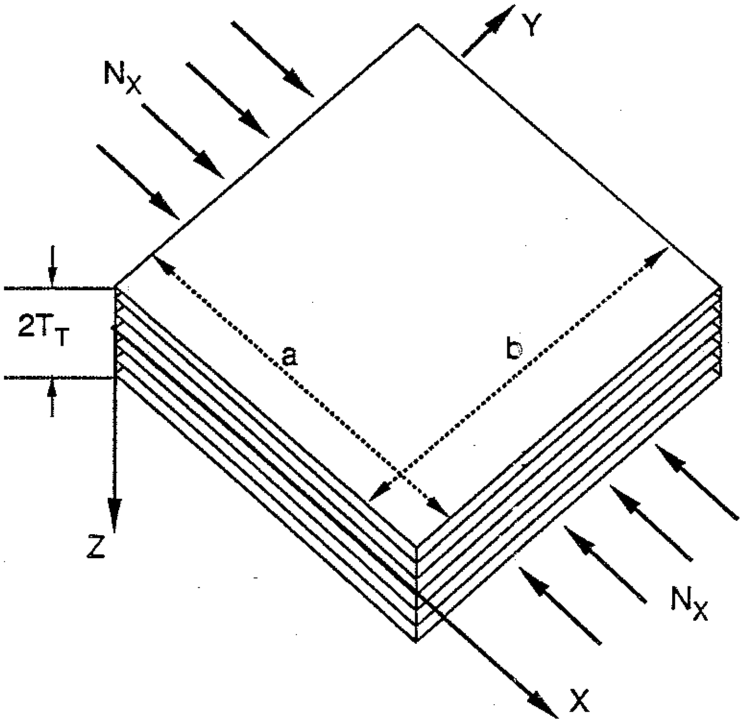

Figure 3. Geometry of composite plate under uniform uniaxial in-plane load.

This is an instance of a general engineering design problem, namely to maximize the lowest buckling load of a structure for a given amount of resources. The structure is discretized by finite elements. Expressing the lowest buckling load with Rayleigh's quotient, the problem is written as

$$\begin{aligned} &\max_{v}\min_{u} \frac{u^T K u}{u^T K_G u} \\ \text{such that} \quad & c^T v - \theta = 0 \\ \text{and} \quad & v_{i\,min} \leq v_i \leq v_{i\,max} \quad \text{for } i = 1, \ldots, M, \end{aligned} \tag{41}$$

where v is a vector of design variables with components v_i, u is the displacement vector, K and K_G (depending on v) are the stiffness matrix and the geometric stiffness matrix, respectively, c is a positive cost vector, and θ is the amount of available resources. The M design variables are subject to upper and lower bounds, $v_{i\,max}$ and $v_{i\,min}$, respectively.

A typical optimization method, applied to solve this problem, starts from a given design and continuously searches for better designs until it finds an optimum design. The trial designs along the path are of no value. The proposed method instead proceeds along a path of optimal designs for increasing amounts of resource θ. The resource θ is varied between the minimum θ_{min} required to satisfy the lower bound constraints and a maximum θ_{max} when all variables are at their upper bounds.

The path consists of several smooth segments, each segment being characterized by a set I_A of variables which are at their upper or lower bounds. Along each segment, some inequality constraints can be treated as equality constraints,

$$v_j = v_{j\,min} \quad \text{or} \quad v_j = v_{j\,max} \quad \text{for } j \in I_A, \tag{42}$$

so that these variables can be eliminated from the optimization problem, while the other variables do not have to be constrained. The optimization problem along a segment can, therefore, be written as

$$\begin{aligned} &\max_{v_i}\min_{u} \frac{u^T K u}{u^T K_G u} \quad \text{for } i \notin I_A \\ \text{such that} \quad & c^T v - \theta = 0. \end{aligned} \tag{43}$$

The solution of the problem consists of three related problems: solving the optimization problem along a segment, locating the end of the segment where the set I_A changes, and finding the set I_A for the next segment.

It is common practice to normalize the displacement vector u such that the denominator of Rayleigh's quotient is unity and to treat this as an equality constraint. Then, using Lagrange multipliers η and μ, the augmented function P^* is formed:

$$P^* = u^T K u - \eta \left[u^T K_G u - 1\right] - \mu \left[c^T v - \theta\right]. \tag{44}$$

The following stationary conditions are obtained by taking the first derivative of P^* with respect to v_i, u, η, and μ, and setting it equal to zero:

i) Optimality conditions

$$u^T \frac{\partial K}{\partial v_i} u - \eta u^T \frac{\partial K_G}{\partial v_i} u - \mu c_i = 0 \quad \text{for } i \notin I_A. \tag{45}$$

ii) Stability conditions

$$Ku - \eta K_G u = 0. \tag{46}$$

iii) Normalization constraint

$$1 - u^T K_G u = 0. \tag{47}$$

iv) Total resource constraint

$$\theta - c^T v = 0. \tag{48}$$

Equations (45)–(48) form a system of nonlinear equations to be solved for v_i, u, η, and μ. A homotopy method is used to find the solutions of these equations as a function of θ.

In certain ranges of structural resources, the optimal solution is known to be bimodal, i.e., the lowest buckling load is a repeated eigenvalue. The formulation for bimodal solutions is given in the appendix of [36]. The existence of bimodal solutions also introduces additional transitions (bimodal to unimodal and vice versa) along the path of optimum solutions.

The homotopy method as described here earlier is intended to solve a *single* nonlinear system of equations, and converge from an arbitrary starting point with probability one. In this context $\theta \in [0,1]$, and the zero curve γ is bounded and leads to the (single) desired solution at $\theta = 1$. The a vector, viewed as an artificial perturbation of the problem, plays a crucial role. In the version of the method employed here, $\theta \in (\theta_0, \theta_1)$, *each* point along γ has physical significance, and a is fixed at zero (no perturbation). Because a is not random, the claimed properties for γ hold only in subintervals (θ_0, θ_1) of $[0, \infty)$. Detecting and dealing with these subinterval transition points is the essence of the modification of the homotopy method used in this section.

Switching from one segment to the next. There are four types of events which end a segment and start a new one:

Type 1: a bound constraint becoming active (i.e., being satisfied as an equality);
Type 2: a bound constraint becoming inactive;
Type 3: transition from a unimodal solution to a bimodal solution;
Type 4: transition from a bimodal solution to a unimodal solution.

To switch from one segment to the next, we first need to locate the transition point. At a transition point there are a number of solution paths which satisfy the stationary equations, and we need to choose the optimum path.

Transition points are located by checking the bound constraints and the optimality conditions. The bound constraints

$$v_{i\,min} \le v_i \le v_{i\,max} \quad \text{for } i = 1, \ldots, M \tag{49}$$

are checked to detect a transition point of type 1.

Optimality of the solution is checked by the Kuhn-Tucker conditions and the second-order conditions discussed below. The solution satisfies the Kuhn-Tucker conditions when all Lagrange multipliers are nonnegative. So a transition of type 2 is detected by checking the positivity of the Lagrange multipliers associated with the bound constraints. These multipliers are obtained by adding the bound constraints to the formulation (43) and replacing the augmented function P^* by

$$P^* = u^T K u - \eta \left[u^T K_G u - 1\right] - \mu \left[c^T v - \theta\right] - \sum_{i \in I_A} \lambda_{1i} \left[v_{i\,min} - v_i\right] - \sum_{i \in I_A} \lambda_{2i} \left[v_i - v_{i\,max}\right]. \quad (50)$$

Taking the first derivative of P^* with respect to v_i gives

$$u^T \frac{\partial K}{\partial v_i} u - \eta u^T \frac{\partial K_G}{\partial v_i} u - \mu c_i + \lambda_{1i} - \lambda_{2i} = 0 \quad \text{for } i \in I_A. \quad (51)$$

Since λ_{1i} is 0 for $v_i \neq v_{i\,min}$ and λ_{2i} is 0 for $v_i \neq v_{i\,max}$ for the above equations, λ_{1i} and λ_{2i} are given by

$$\begin{aligned} \lambda_{1i} &= -u^T \frac{\partial K}{\partial v_i} u + \eta u^T \frac{\partial K_G}{\partial v_i} u + \mu c_i \quad \text{for } v_i = v_{i\,min} \\ \lambda_{2i} &= u^T \frac{\partial K}{\partial v_i} u - \eta u^T \frac{\partial K_G}{\partial v_i} u - \mu c_i \quad \text{for } v_i = v_{i\,max}. \end{aligned} \quad (52)$$

A type 2 transition is detected by a Lagrange multiplier becoming nonpositive. Similar equations for the bimodal case are given in the appendix of [36].

The bimodal formulation replaces η by η_1 and η_2 which are the Lagrange multipliers for the normalization constraints on the two buckling modes. When one of them becomes negative, the corresponding mode should be removed for the optimum design, so that we have a transition of type 4 from bimodal to unimodal design.

For a transition of type 3, we need to check if there is another buckling mode associated with a lower buckling load. This can be accomplished by checking the second-order optimality conditions for the buckling mode variables u given by

$$r^T \left[\nabla_u^2 P^*\right] r > 0 \quad \text{for every } r \text{ such that} \quad \nabla_u h^T r = 0 \quad (53)$$

where

$$\begin{aligned} \left[\nabla_u^2 P^*\right] &= \left[\frac{\partial^2 P^*}{\partial u_s \partial u_t}\right] \\ \nabla_u h &= \left[\frac{\partial h}{\partial u_s}\right] \\ h &= u^T K_G u - 1. \end{aligned}$$

Alternatively we can solve the buckling problem (46) for the current design and check whether the buckling load obtained from the stationary conditions is truly the lowest one. The transition of type 3 is detected by checking if

$$p \neq p_1 \quad (54)$$

where p is the buckling load obtained from the stationary conditions while p_1 is the first buckling load obtained by solving the stability conditions (46) for the given structure.

Once a transition point is located, we need to choose a path which satisfies the optimality conditions. Choosing an optimum path constitutes finding a set of active bound constraints for type 1 and 2 transitions and the correct buckling modes for type 3 and 4 transitions. These are obtained by using the Lagrange multipliers of the previous path and the sensitivity calculation on the buckling load. The procedure is explained separately for each type of transition.

A type 1 transition occurs when one of design variables, v_i, hits the upper or lower bound. Then v_i is set at $v_{i\,max}$ or $v_{i\,min}$ and treated as a constant value. The number of design variables is reduced by one.

At a type 2 transition, one of the Lagrange multipliers for the bound constraints, λ_{1i} and λ_{2i}, is found to be negative. The bound constraint corresponding to the negative λ_{1i} or λ_{2i} is set to be inactive and the number of design variables is increased by one.

At a transition from a unimodal solution to a bimodal solution (a type 3 transition), the formulation requires two buckling modes, u_1 and u_2, for the solution of the upcoming bimodal path. These modes can be obtained by solving the stability conditions (46) of the previous unimodal formulation, since the stability conditions give two buckling modes at the bimodal transition point.

At a transition from a bimodal to a unimodal solution (a type 4 transition), two buckling modes are given from the bimodal solution. One of the Lagrange multipliers for the normalization constraints, η, is known to be negative from the previous transition check, so the buckling mode corresponding to the positive η is chosen.

Some of the above transitions can occur simultaneously. Special treatment is required in certain cases where the Lagrange multipliers are not available. In general, the optimum design requires at least one design variable v_i for a unimodal case and two design variables for a bimodal case. At a type 1 transition, the number of design variables is reduced by one, and at a type 3 transition the bimodal formulation requires one more design variable in case the previous unimodal path has only one design variable. So some type 1 or type 3 transitions occur simultaneously with a type 2 transition which allows an additional design variable. In that case, the Lagrange multipliers λ_{1i} and λ_{2i}, which are used at a type 2 transition to determine a new design variable, are not available. We then rely on the sensitivity information of p with respect to v. For a unimodal case, the location of the new design variable v_i is determined where $dp/d\theta$ is maximized. For a bimodal case, we need to find a combination of i and j which maximizes the value of the bimodal buckling load for a small increment of the total available resource. Considering the bound constraints in the formulation, the new design variables are determined by

$$\max_{i,j} \frac{dp}{d\theta} = \frac{\partial p_1}{\partial v_i}\frac{dv_i}{d\theta} + \frac{\partial p_1}{\partial v_j}\frac{dv_j}{d\theta} \tag{55}$$

such that

$$\begin{aligned}\frac{\partial p_1}{\partial v_i}\frac{dv_i}{d\theta}+\frac{\partial p_1}{\partial v_j}\frac{dv_j}{d\theta}&=\frac{\partial p_2}{\partial v_i}\frac{dv_i}{d\theta}+\frac{\partial p_2}{\partial v_j}\frac{dv_j}{d\theta}\\ \frac{dv_i}{d\theta}&\geq 0 \quad \text{for } v_i = v_{i\,min}\\ \frac{dv_i}{d\theta}&\leq 0 \quad \text{for } v_i = v_{i\,max}\\ \frac{dv_j}{d\theta}&\geq 0 \quad \text{for } v_j = v_{j\,min}\\ \text{and } \frac{dv_j}{d\theta}&\leq 0 \quad \text{for } v_j = v_{j\,max}\end{aligned}$$

where p_1 and p_2 are the buckling loads corresponding to the buckling modes u_1 and u_2, respectively.

After we obtain the design variables v and the buckling modes u, we need the Lagrange multipliers μ and η at the transition point to complete the set of starting values for the next solution path. These are obtained by solving the stationary conditions for the given u and v. For example, in the unimodal case, η is obtained from the stability conditions (46) and μ is obtained by solving one of the optimality conditions (45).

Summary. A typical optimization method starts from a given design and continuously searches for better designs until it finds an optimum design. The trial designs along the path are of no value. Here a strategy for tracing a path of optimum solutions parameterized by an amount of available resources was discussed. Equations for the optimum path were obtained using Lagrange multipliers, and were solved by a homotopy method.

The solution path has several branches due to changes in the active constraint set and transition from unimodal to bimodal solutions. The Lagrange multipliers and the second-order optimality conditions were used to detect branching points and to switch to the optimum solution path.

In [36] this procedure was applied to the design of a foundation which supports a column for maximum buckling load, where the total available foundation was used as a homotopy parameter. Starting from a minimum foundation which satisfies the lower bound (in this case zero), a set of optimum foundation designs was obtained for the full range of total foundation stiffness. Numerical results for the design of composite plates described here, where the total plate thickness is the resource parameter being varied, are in [37].

9. Kreisselmeier-Steinhauser envelope function. The previous sections presented ways that were both theoretically "correct" and computationally "practical" to deal with inequality constraints. However, there are numerous practical difficulties in those approaches, and the implementation and tuning details become absolutely crucial. For example, with the expanded Lagrangian formulation, line searches may generate negative arguments for the ln functions, and the homotopy zero curve may diverge if the Step 1 solution is not good enough. For the active set approach, the detection and switching criteria for transition points may become extremely cumbersome and inefficient. This section suggests an alternate way of dealing with inequality constraints.

Consider inequality constraints of the form

$$g_i(x) \le 0, \quad i = 1, \ldots, m, \tag{56}$$

where each $g_i : E^n \to E$ is C^2. For a constant $\rho > 0$, the Kreisselmeier-Steinhauser [14] envelope function for (56) is

$$K(x) = \frac{1}{\rho} \ln \left[\sum_{i=1}^{m} \exp\big(\rho g_i(x)\big) \right]. \tag{57}$$

$K(x)$ is a cumulative measure of the satisfaction or violation of the constraints (56). Let $g_{max}(x) = \max\{g_1(x), \ldots, g_m(x)\}$, and observe that

$$K(x) = g_{max}(x) + \frac{1}{\rho} \ln \left[\sum_{i=1}^{m} \exp\Big(\rho\big(g_i(x) - g_{max}(x)\big)\Big) \right], \tag{58}$$

from which it directly follows that

$$g_{max}(x) \le K(x) \le g_{max}(x) + \frac{1}{\rho} \ln m. \tag{59}$$

Thus the envelope $K(x)$ follows the maximum constraint, more closely for large ρ. In particular, (56) could be replaced by

$$K(x) \le 0 \tag{60}$$

with an error of no more than $(\ln m)/\rho$.

The choice of ρ involves a tradeoff between modelling the maximum constraint (large ρ preferred) and avoiding large gradients (small ρ preferred). If the practical criterion for an active constraint is $|g_i| \le \epsilon$, then a choice for ρ which has worked well in practice is

$$\rho = \frac{\ln m}{\epsilon}. \tag{61}$$

Observe that $K(x)$ is C^2 and defined *everywhere*, a decided advantage over barrier functions. Furthermore, (60) is a *single* nonlinear constraint, which makes any active set strategy very simple. (60) has been successfully used in large scale structural optimization [4] and optimal control [14].

10. Probability-one homotopy for Kuhn-Tucker optimality conditions. The approaches of earlier sections are still not always entirely adequate. The cumulative constraint function (57) is decidedly unnatural, extremely nonlinear and ill conditioned for large ρ, and does not take advantage of a known solution to a related problem. Consider again the general nonlinear programming problem:

$$\begin{aligned} &\min \theta(x) \\ \text{subject to} \quad &g(x) \le 0, \\ &h(x) = 0, \end{aligned} \tag{62}$$

under the same assumptions mentioned before. The Kuhn-Tucker necessary optimality conditions for (62) are

$$\begin{aligned}\nabla\theta(x)+\beta^t\nabla h(x)+\mu^t\nabla g(x)&=0,\\ h(x)&=0,\\ g(x)&\le 0,\\ \mu&\ge 0,\\ \mu^t g(x)&=0,\end{aligned}\tag{63}$$

where $\beta\in E^p$ and $\mu\in E^m$. Following Mangasarian [19] and Watson [40], the complementarity conditions $\mu\ge 0$, $g(x)\le 0$, $\mu^t g(x)=0$ are replaced by the equivalent nonlinear system of equations

$$W(x,\mu)=0, \tag{64a}$$

where

$$W_i(x,\mu)=-\left|\mu_i+g_i(x)\right|^3+\mu_i^3-\left(g_i(x)\right)^3,\quad i=1,\ldots,m. \tag{64b}$$

Thus the optimality conditions (63) take the form

$$F(x,\beta,\mu)=\begin{pmatrix}[\nabla\theta(x)+\beta^t\nabla h(x)+\mu^t\nabla g(x)]^t\\ h(x)\\ W(x,\mu)\end{pmatrix}=0. \tag{65}$$

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

$$\rho_a(\lambda,z)=\lambda F(z)+(1-\lambda)(z-a), \tag{66}$$

where $a\in E^{n+p+m}$. Simple conditions on θ, g, and h guaranteeing that the above homotopy map $\rho_a(\lambda,z)$ will work are unknown, although this map has worked very well on some difficult fuel optimal orbital rendezvous problems [38].

Frequently in practice the functions θ, g, and h involve a parameter vector c, and a solution to (62) is known for some $c=c^{(0)}$. Suppose that the problem under consideration has parameter vector $c=c^{(1)}$. Then

$$c=(1-\lambda)c^{(0)}+\lambda c^{(1)} \tag{67}$$

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 (65) become functions of λ as well, $F(\lambda,x,\beta,\mu)=0$, and

$$\rho_a(\lambda,z)=\lambda F(\lambda,z)+(1-\lambda)(z-a) \tag{68}$$

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

$$F(\lambda,z)=0, \tag{69}$$

since the solution $z^{(0)}$ to $F(0,z)=0$ (the problem corresponding to $c=c^{(0)}$) is known. However, for various technical reasons, (68) is much better than (69) [38].

11. Fuel-optimal orbital rendezvous problem. The problem is to find a minimum fuel rendezvous trajectory between two bodies, the non-maneuvering 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). A final impulse is applied at the end of the interceptor trajectory to provide a velocity match with the target. Hence the number of impulses equals the number of coasting arcs. The maneuver must be completed within some specified time and the trajectory must avoid passing too near 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 notation used is:

η	– change in true anomaly,
$\vec{r}(\eta)$	– radius vector,
$\hat{r}(\eta)$	– unit vector in the radial direction,
u	– reciprocal of the magnitude of the radius vector,
$\vec{v}(\eta)$	– velocity vector,
$h(\eta)$	– magnitude of the angular momentum vector,
$\hat{h}(\eta)$	– unit vector in the direction of angular momentum.

The variables are the coasting angles on each arc including a possible initial coast, the components of the velocity change vector, and the coasting angle of the target. The forward equations of motion for any subarc are:

$$u(\eta) = \frac{\mu}{h^2} + \left(u(0) - \frac{\mu}{h^2}\right)\cos\eta + u'(0)\sin\eta,$$
$$\hat{r}(\eta) = \hat{r}(0)\cos\eta + \hat{r}'(0)\sin\eta,$$

with time of flight

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

The constraints are:

final position match	$\vec{r}_f - \vec{r}_t(\eta_t) = 0,$
final velocity match	$\vec{v}_f - \vec{v}_t(\eta_t) = 0,$
time of flight constraint	$T_f - T_t = 0,$
nonnegativity of the coasting arcs of the interceptor	$\eta_i \geq 0, \quad i = 1, \ldots, nim,$
nonnegativity of the coasting arc of the target	$\eta_t \geq 0,$
time limit for rendezvous	$T_{\max} - T_f \geq 0,$

minimum radius constraint for each coasting arc...

except the initial coast arc of the interceptor.. $u_0 - u_{j\,\max} \geq 0, \quad j = 1, \ldots, nim - 1,$

nonnegativity of the radius constraint .... $u_{j\,\min} \geq 0, \quad j = 1, \ldots, nim - 1.$

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. nim is the number of impulses. The value of $u_{j\,\max}$ in these constraints is given by the rather awkward and difficult to compute expression

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

The optimization problem, subject to all the above constraints, is

$$\min_S V(x),$$

where

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

and

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

For u, u', and h, the subscript j denotes the conditions at the beginning of the jth subarc, and on the variables $\Delta u'$, Δh, and ϕ the subscript j denotes the jth impulse which occurs at the end of the jth subarc.

Using the formulation of equations (65) and (68), numerous such rendezvous problems have been solved, both in-plane and out-of-plane, and with 2, 3, 4, or 5 impulses. See [38] for more details.

12. Global optimization of polynomial programs. Let E^n denote n-dimensional real Euclidean space, C^n denote n-dimensional complex Euclidean space, $E^{m\times n}$ the set of real $m \times n$ matrices, and $C^{m\times n}$ the set of complex $m \times n$ matrices. Lower case Greek letters will be real or complex scalars or scalar-valued functions, and Roman letters will generally denote vectors and vector-valued functions. x_i denotes the ith component of the vector x, and for a matrix A, A_{ij} denotes the i,j entry, $A_{\cdot j}$ denotes the jth column, and $A_{i\cdot}$ denotes the ith row. The Jacobian matrix of a function $f(x)$ is written $\nabla f(x)$.

Consider the general nonlinear programming problem

$$\begin{aligned} &\min \theta(x) \\ \text{subject to } &g(x) \leq 0, \\ &h(x) = 0, \end{aligned} \tag{70}$$

where $\theta : E^n \to E$, $g : E^n \to E^p$, and $h : E^n \to E^q$ are polynomial functions. Precisely, each component of θ, g, and h has the form

$$\sum_{k=1}^{n_i} \alpha_{ik} \prod_{j=1}^{n} x_j^{d_{ijk}},$$

where the α_{ik} are real and the d_{ijk} are nonnegative integers. Such a problem will be called a *polynomial programming problem.*

By adding slack variables, the inequality constraints

$$g_k(x) \leq 0$$

can be converted to equality constraints

$$g_k(x) + y_k^2 = 0.$$

Henceforth, assume that all the constraints are equality constraints, so that (70) becomes

$$\begin{aligned} &\min \theta(x) \\ &\text{subject to} \quad h(x) = 0. \end{aligned} \tag{71}$$

The Lagrange Multiplier Theorem says that if $\bar{x}$ is a locally optimal solution for (71) and rank $\nabla h(\bar{x}) = q$, then $\exists\, \bar{r} \in E^q$ such that

$$\begin{aligned} \nabla\theta(\bar{x}) + \bar{r}^t \nabla h(\bar{x}) &= 0, \\ h(\bar{x}) &= 0. \end{aligned} \tag{72}$$

Since θ and h are polynomial functions, so are $\nabla\theta$ and ∇h. Thus (72) is a polynomial system of $n + q$ equations in the $n + q$ unknowns x and r. So the *polynomial programming problem* (70) reduces to the *polynomial system of equations* (72).

Polynomial continuation provides a globally convergent homotopy algorithm guaranteed to find *all* the solutions of (72), and thus the global optimum of (70). In [36], [37], [40], [47], [49], [53], [54] homotopy methods are used to solve optimization problems, but only for *local* (not *global*) optimization. In [12] and [13] traditional polynomial continuation is used to solve the global polynomial programming problem. In this paper, the more efficient m-homogeneous theory is applied. The rest of this section updates traditional polynomial continuation theory to the m-homogeneous context, develops a standard m-homogeneous formulation of (72), and illustrates the theory with several examples.

m-homogeneous polynomial continuation. Polynomial continuation is a numerical method for computing all the geometrically isolated solutions to polynomial systems. Let $f(z) = 0$ denote a system of N polynomial equations in N unknowns. The degree of the i^{th} equation is $d_i = \max_k \sum_{j=1}^{N} d_{ijk}$ and

$$td = \prod_{i=1}^{N} d_i$$

is the *total degree* of the system. Traditional polynomial continuation computes the full list of geometrically isolated solutions to $f(z) = 0$ by numerically tracking td paths in the space $C^N \times [0,1]$. See, for example, [24] and [46].

Although this method works quite well when td is relatively small, the computational cost for larger systems can be prohibitive. A recent advance in polynomial continuation, the m-homogeneous approach of Morgan and Sommese [25]–[27], reduces the number of paths that must be tracked in many cases. By partitioning the variables to create an *m-homogeneous structure*, we can solve the system tracking only the *Bezout number* of paths. Frequently, we can arrange for the Bezout number to be smaller than the total degree. The mechanics of numerically tracking the paths is essentially the same as for the traditional method. Here, we describe how to create an m-homogeneous structure and find the Bezout number. In the Appendix, the method and its theory are more fully developed.

We create an m-homogeneous structure for $f(z)$ by partitioning the variables $z_1, z_2, \ldots, z_N$ into m nonempty sets. It will simpler for the exposition if we re-index with double subscripts. Thus

$$\{z_1, \ldots, z_N\} = \bigcup_{j=1}^{m} \{z_{1,j}, \ldots, z_{k_j,j}\},$$

where $\sum_{j=1}^{m} k_j = N$. Now choose homogeneous variables $z_{0,j}$ for $j = 1$ to m and define

$$Z_j = \{z_{0,j}, z_{1,j}, \ldots, z_{k_j,j}\}$$

for $j = 1$ to m. Then evoke the substitution $z_{i,j} \leftarrow z_{i,j}/z_{0,j}$ for $i = 1$ to k_j and $j = 1$ to m, generating a system $f' = 0$ of N equations in $N + m$ unknowns (after clearing the denominators of powers of the $z_{0,j}$). This f' is called *m-homogeneous* because the variables are partitioned into m collections, $Z_1, \ldots, Z_m$, so that f' is homogeneous as a system in the variables of any one of the collections. We take $d_{j,l}$ to denote the j^{th} degree of the l^{th} polynomial; that is, with all variables held fixed except those in Z_j, f'_l has homogeneous degree $d_{j,l}$. Polynomial f'_l is said to have *type* $= (d_{1,l}, \ldots, d_{m,l})$.

The *Bezout number* d of an m-homogeneous polynomial system is given by

$$d = \text{Coef}\left[D, \prod_{j=1}^{m} \phi_j^{k_j}\right] \tag{73}$$

where

$$D = \prod_{l=1}^{n} \sum_{j=1}^{m} d_{j,l}\phi_j, \tag{74}$$

i.e., d is the coefficient of the $\prod_{j=1}^{m} \phi_j^{k_j}$ term of D. Frequently, an m-homogenization of f for $m > 1$ has a (much) smaller Bezout number than the $m = 1$ case, where d equals the total degree of f.

Example 1. Consider the following system:

$$\begin{aligned} z_1 z_2 z_3 z_4 + 1 &= 0, \\ z_1 z_3 + z_2 z_4 + z_1 z_4 &= 0, \\ 4z_1 z_3 z_4 - 2z_2 z_3 z_4 + 1 &= 0, \\ z_1 + z_2 &= 0. \end{aligned} \tag{75}$$

By grouping the variables of (75) into different sets, we create different m-homogeneous structures and Bezout numbers. Normally, we would want to solve such a system with the m-homogeneous structure that gives the smallest Bezout number. For each grouping of variables, we will form the combinatorial product D defined in (74) above, and then pick out the distinguished coefficient that gives the Bezout number d. Thus:

Example 1.1. Group variables as: $\{z_1, z_2\} \cup \{z_3, z_4\}$. Then, $D \equiv (2\phi_1 + 2\phi_2)(\phi_1 + \phi_2)(\phi_1 + 2\phi_2)(\phi_1 + 0\phi_2)$ and $d = \text{Coef}\,[D, \phi_1^2\phi_2^2]$, Thus, $d = 10$.

Example 1.2. Group variables as: $\{z_1, z_2\} \cup \{z_3\} \cup \{z_4\}$. Then, $D = (2\phi_1 + \phi_2 + \phi_3)(\phi_1 + \phi_2 + \phi_3)^2(\phi_1 + 0\phi_2 + 0\phi_3)$, and $d = \text{Coef}\,[D, \phi_1^2\phi_2\phi_3] = 8$.

Example 1.3. Group variables as: $\{z_1\} \cup \{z_2\} \cup \{z_3, z_4\}$. Then, $D = (\phi_1 + \phi_2 + 2\phi_3)^2(\phi_1 + \phi_2 + \phi_3)(\phi_1 + \phi_2 + 0\phi_3)$, and $d = \text{Coef}\,[D, \phi_1\phi_2\phi_3^2] = 16$.

We see that 1.2 gives the smallest Bezout number. Thus, while the 1-homogeneous (traditional) polynomial continuation yields a 24 path homotopy (i.e., the number of the total degree), we can (easily) find a 3-homogeneous 8 path homotopy. Such a savings in computer work (i.e., by a factor of 1/3) can be significant in some applications.

To summarize: given a system of N polynomial equations in N unknowns, there exist traditional constructions from polynomial continuation that yield the total-degree number of paths to track in order to compute the complete list of geometrically isolated solutions. The purpose of the m-homogeneous approach is to reduce the number of paths needed to solve the problem, thereby realizing a savings in computational work. The m-homogeneous method is begun by partitioning the variables into sets. Then (73) yields the Bezout number, the associated number of paths to be tracked. A different partitioning of the variables yields a different Bezout number, and there is no systematic way to find the partitioning that yields the smallest Bezout number, aside from exhaustive search.

An m-homogeneous homotopy formulation of the polynomial program. This section gives a partitioning of the variables of the polynomial programming problem and presents a simple formula for the associated Bezout number. It should be noted that for any particular problem, additional savings can often be obtained by customized the m-homogeneous partitioning to the problem. Specific examples are given in the next section.

Partition the variables into two sets S_1 and S_2, where

$$S_1 = \{x_1, x_2, \ldots, x_n\} \text{ and } S_2 = \{r_1, r_2, \ldots, r_q\},$$

corresponding to a 2-homogeneous structure.

Consider the i^{th} Lagrangian equation

$$(\nabla_x L)_i = \nabla\theta_i + \sum_{j=1}^{q} r_j \nabla h_{j,i}$$

where $L(x,r) = \theta(x) + r^t h(x)$ is the Lagrangian function for (71) and

$$\nabla h_{j,i} = \frac{\partial h_j}{\partial x_i}.$$

Define, for $i = 1$ to n,

$$\delta_i^{\nabla\theta} = \begin{cases} -1, & \text{if } \nabla\theta_i \equiv 0, \\ \deg(\nabla\theta_i), & \text{otherwise,} \end{cases}$$

and

$$\delta_i^{\nabla h} = \begin{cases} -1, & \text{if } \nabla h_{j,i} \equiv 0 \text{ for } j = 1 \text{ to } q, \\ \max\{\deg(\nabla h_{j,i}) \mid j = 1 \text{ to } q\}, & \text{otherwise.} \end{cases}$$

By re-ordering the indices of the x_i if necessary, there is a nonnegative integer $q_0 \leq q$ so that $\delta_i^{\nabla h} = -1$ for $i = 1$ to q_0, and $\delta_i^{\nabla h} \neq -1$ for $i = q_0 + 1$ to q. Take $q_0 = 0$ if no $\delta_i^{\nabla h} = -1$. Note that, by the definition of q_0, x_1, x_2, …, x_{q_0} are the variables that do not appear in h and x_{q_0+1}, …, x_n are the ones that do.

Define

$$\delta_i = \max\{\delta_i^{\nabla\theta}, \delta_i^{\nabla h}\} \qquad \text{for } i = 1 \text{ to } n,$$

and

$$\delta_i^h = \deg(h_i) \qquad \text{for } i = 1 \text{ to } q.$$

Assume (without loss of generality) that $\delta_i \neq -1$, $\delta_i^h \neq 0$, and, if $\delta_i^{\nabla\theta} = 0$, then $\delta_i^{\nabla h} \neq -1$, for any i.

We can now generate the Bezout number. The type of the i^{th} Lagrangian equation is $(\delta_i, 1)$ if $i > q_0$ and $(\delta_i, 0)$ if $i \leq q_0$. The type of the i^{th} constraint equation is $(\delta_i^h, 0)$. Then D from equation (74) is defined from these types as

$$D = \prod_{i=1}^{q} \delta_i^h \phi_1 \prod_{i=1}^{q_0} \delta_i \phi_1 \prod_{i=q_0+1}^{n} (\delta_i \phi_1 + \phi_2),$$

where the second factor is omitted if $q_0 = 0$, and ϕ_1 and ϕ_2 correspond to S_1 and S_2, respectively. Then the Bezout number is given by

$$d = \text{Coef}\left[D, \phi_1^n \phi_2^q\right].$$

Note that we will consider only the case $q < n$. If $q > n$, then either the constraint set cannot be satisfied or it can be reduced by omitting redundant equations. If $q = n$, then either the constraint set can be solved as an independent system or it can be reduced by omitting redundant equations. The usual rank $\nabla h(\bar{x}) = q$ condition rules out these redundant cases.

We see that

$$d = \prod_{i=1}^{q} \delta_i^h \prod_{i=1}^{q_0} \delta_i \, \text{Coef}\left[D', \phi_1^{n-q_0-q} \phi_2^q\right]$$

where

$$D' = \prod_{i=q_0+1}^{n} (\delta_i \phi_1 + \phi_2),$$

and, by some simple combinatorial observations, we conclude that

$$d = \prod_{i=1}^{q} \delta_i^h \prod_{i=1}^{q_0} \delta_i \sum_{q_0+1 \le i_1 < i_2 < \cdots < i_{n-q_0-q} \le n} \delta_{i_1} \delta_{i_2} \cdots \delta_{i_{n-q_0-q}}. \tag{76}$$

By comparison, the total degree is

$$td = \prod_{i=1}^{q} \delta_i^h \prod_{i=1}^{n} \max\left\{\delta_i^{\nabla \theta}, \delta_i^{\nabla h} + 1\right\}.$$

For example, if $q = n - 1$ (e.g., the number of constraints is one less than the number of variables) and $q_0 = 0$, then we get

$$d = \prod_{i=1}^{q} \delta_i^h \sum_{i=1}^{n} \delta_i.$$

Another case of interest occurs when the objective function and constraints are all quadratics. For simplicity, assume each variable occurs in at least one constraint raised to the second power. (The resulting Bezout number will be an upper bound for the other quadratic cases.) Then $q_0 = 0$ and $\delta_i^{\nabla h} = 1$ for all i. It follows that

$$\begin{aligned} d &= 2^q \sum_{1 \le i_1 < \cdots < i_{n-q} \le n} 1 \\ &= 2^q \binom{n}{n-q} \\ &= 2^q \binom{n}{q}. \end{aligned} \tag{77}$$

Compare the generally much larger total degree in this case:

$$td = 2^{q+n}.$$

Examples. This section presents two specific examples. The first is a realistic "small" problem that arises in geometric modeling. The second is a prototype structural design problem.

Geometric modeling problem. Let P_1 and P_2 be two polynomial surfaces in E^3. The problem is to compute the distance between P_1 and P_2, defined to be the length of the smallest line segment connecting them. Assuming that

$$P_i = \{z \in E^3 \mid h_i(z) = 0\}$$

where h_i is a d_i^{th} degree polynomial, the problem becomes

$$\begin{aligned} \min \ & \theta(x,y) = \|x - y\|^2 \\ \text{subject to} \ \ & h_1(x) = 0, \\ & h_2(y) = 0. \end{aligned}$$

The necessary conditions (72) in this case are

$$\begin{aligned} \nabla_{(x,y)}L(x,y,r) = 2\big[&x_1 - y_1, x_2 - y_2, x_3 - y_3, -(x_1 - y_1), -(x_2 - y_2), -(x_3 - y_3)\big] \\ &+ r_1\big[\nabla h_{1,1}, \nabla h_{1,2}, \nabla h_{1,3}, 0, 0, 0\big] \\ &+ r_2\big[0, 0, 0, \nabla h_{2,4}, \nabla h_{2,5}, \nabla h_{2,6}\big], \\ h_1(x) = 0,& \\ h_2(y) = 0.& \end{aligned}$$

Here $n = 6$, $q = 2$, and $\delta_i^{\nabla\theta} = 1$ for $i = 1$ to 6. Then

$$\begin{aligned} \delta_i^{\nabla h} &= \begin{cases} -1, & \text{if } \nabla h_{1,i} \equiv 0, \\ \deg(\nabla h_{1,i}), & \text{otherwise,} \end{cases} \qquad \text{for } i = 1 \text{ to } 3, \\ \delta_i^{\nabla h} &= \begin{cases} -1, & \text{if } \nabla h_{2,i} \equiv 0, \\ \deg(\nabla h_{2,i}), & \text{otherwise,} \end{cases} \qquad \text{for } i = 4 \text{ to } 6, \\ \delta_i &= \max\{1, \delta_i^{\nabla h}\} \qquad \text{for } i = 1 \text{ to } 6, \end{aligned}$$

and

$$\delta_i^h = d_i \qquad \text{for } i = 1, 2.$$

For the special case $q_0 = 0$, equation (76) gives

$$d = \delta_1^h \delta_2^h \sum_{1 \le i_1 < i_2 < i_3 < i_4 \le 6} \delta_{i_1}\delta_{i_2}\delta_{i_3}\delta_{i_4}.$$

If $q_0 = 0$ and h_1 and h_2 are quadratics, then $\delta_i = 1$ for all i, and by (77)

$$d = 4\binom{6}{2} = 60,$$

while the total degree is $2^8 = 256$.

However, one can do better with a customized m-homogeneous breakdown; namely, partition the variables as

$$\{x_1, x_2, x_3, y_1, y_2, y_3\} \cup \{r_1\} \cup \{r_2\}.$$

Then, for $q_0 = 0$, (74) and (73) become

$$D = \delta_1^h \phi_1 \delta_2^h \phi_1 \prod_{i=1}^{3} (\delta_i \phi_1 + \phi_2) \prod_{i=4}^{6} (\delta_i \phi_1 + \phi_3)$$

and

$$d = \text{Coef}\left[D, \phi_1^6 \phi_2 \phi_3\right].$$

If in addition the h_i are quadratics, then

$$D = 2\phi_1 2\phi_1 (\phi_1 + \phi_2)^3 (\phi_1 + \phi_3)^3$$

and

$$d = \text{Coef}\left[D, \phi_1^6 \phi_2 \phi_3\right] = 36.$$

Thus, the customized m-homogeneous structure reduces the Bezout number for the case $q_0 = 0$ and h_i quadratic from 60 to 36.

If, in fact, $q_0 \neq 0$, then further reductions are possible. Consider the case that P_1 is a cylinder and P_2 is a sphere, as follows:

$$h_1(x) = x_2^2 + x_3^2 - 1,$$
$$h_2(y) = y_1^2 + (y_2 - 3)^2 + y_3^2 - 1.$$

The variable x_1 does not appear in the constraint set. Therefore $q_0 = 1$, and

$$D = 2\phi_1 2\phi_1 \ \phi_1 (\phi_1 + \phi_2)^2 (\phi_1 + \phi_3)^3$$

and

$$d = \text{Coef}\left[D, \phi_1^6 \phi_2 \phi_3\right] = 24.$$

Going back to the standard 2-homogeneous Bezout number, equation (76) yields

$$d = 2^2 1 \binom{6-1}{6-1-2} = 40.$$

Thus, the customized m-homogeneous approach is better, but in both cases it is better to exploit $q_0 = 1$.

Structural design problem. Let us consider the following prototype structural design problem from [53]:

$$\begin{aligned} &\min c_1x_1 + \cdots + c_{2k}x_{2k} \\ \text{subject to } \quad &x_{2i-1}^2 + x_{2i}^2 - b_i \le 0, \qquad i = 1 \text{ to } k, \\ &\sum_{j=1}^{2k} a_{i,j}x_j = 0, \qquad i = 1 \text{ to } s, \end{aligned}$$

where c_i, $b_i > 0$, $a_{i,j}$ are constants and k and s are positive integers with $s \le 2k$.

Introducing slack variables and our standard notation gives

$$\begin{aligned} n = 3k, \quad q &= k + s, \quad q_0 = 0, \\ \theta(x) &= c_1x_1 + \cdots + c_{2k}x_{2k}, \\ h_i(x) &= x_{2i-1}^2 + x_{2i}^2 + x_{2k+i}^2 - b_i, \qquad i = 1 \text{ to } k, \\ h_{k+i}(x) &= \sum_{j=1}^{2k} a_{i,j}x_j, \qquad i = 1 \text{ to } s, \end{aligned}$$

and thus

$$\begin{aligned} (\nabla_x L)_{2i-1} &= c_{2i-1} + 2r_ix_{2i-1} + \sum_{l=1}^{s} r_{k+l}a_{l,2i-1} \\ (\nabla_x L)_{2i} &= c_{2i} + 2r_ix_{2i} + \sum_{l=1}^{s} r_{k+l}a_{l,2i} \\ (\nabla_x L)_{2k+i} &= 2r_ix_{2k+i} \end{aligned}$$

for $i = 1$ to k.

Then

$$\begin{aligned} \delta_i^{\nabla\theta} &= 0 \text{ or } -1, \qquad i = 1 \text{ to } n, \\ \delta_i^{\nabla h} &= 1, \qquad i = 1 \text{ to } n, \\ \delta_i &= 1, \qquad i = 1 \text{ to } n, \\ \delta_i^h &= 2, \qquad i = 1 \text{ to } k, \\ \delta_i^h &= 1, \qquad i = k+1 \text{ to } k+s. \end{aligned}$$

Thus by (76)

$$\begin{aligned} d &= 2^k 1^s \sum_{1 \le i_1 < \cdots < i_{2k-s} \le n} \delta_1 \cdots \delta_{2k-s} \\ &= 2^k \binom{3k}{2k-s} \\ &= 2^k \binom{3k}{k+s}. \end{aligned}$$

Table 1. Total degree, standard Bezout number, and customized Bezout number for the prototype structural design problem.

k	s	total degree	standard Bezout	customized Bezout
4	1	65536	12672	6912
4	2	65536	14784	29376
4	3	65536	12672	94462
4	4	65536	7920	214080
4	5	65536	3520	314880
4	6	65536	1056	293760
4	7	65536	192	161280

(Compare equation (77).) The total degree, by contrast, is

$$td = 2^{4k}.$$

Sometimes, one can do better with a customized Bezout breakdown. Consider the partitioning of variables:

$$\bigcup_{i=1}^{k}\{x_{2i-1}, x_{2i}, x_{2k+i}\} \bigcup_{i=1}^{k}\{r_i\} \bigcup\{r_{k+1}, \ldots, r_{r+s}\}.$$

To compute the combinatorial product D, assign dummy variables $\phi_1, \phi_2, \ldots, \phi_k$ to the first k groups, $\phi_{k+1}, \phi_{k+2}, \ldots, \phi_{2k}$ to the second k groups, and ϕ_{2k+1} to the last group. Then

$$D = 2^k \phi_1 \cdots \phi_k (\phi_1 + \cdots + \phi_k)^s \prod_{i=1}^{k} (\phi_i + \phi_{k+i} + \phi_{2k+1})^2 (\phi_i + \phi_{k+i})$$

and

$$d = \text{Coef}\left[D, \left(\prod_{i=1}^{k} \phi_i^3 \phi_{k+i}\right) \phi_{2k+1}^s\right].$$

This simplifies to

$$d = 2^k \text{Coef}\left[D', \left(\prod_{i=1}^{k} \phi_i^2 \phi_{k+i}\right) \phi_{2k+1}^s\right]$$

where

$$D' = \left[\prod_{i=1}^{k} 3\phi_i^2 \phi_{k+i} + 2\phi_i(\phi_i + 2\phi_{k+i})\phi_{2k+1} + (\phi_i + \phi_{k+i})\phi_{2k+1}^2\right] (\phi_1 + \cdots + \phi_k)^s.$$

Table 1 gives the total degree, standard Bezout number, and customized Bezout number for the case $k = 4$. Note that the customized Bezout number is better than the standard Bezout number for $s = 1$ but worse for $s > 1$.

13. Acknowledgement. The author is indebted to all those who have collaborated on aspects of this work, in particular, J. Patrick Bixler, Raphael T. Haftka, Manohar P. Kamat, Frederick H. Lutze, Alexander P. Morgan, Katta G. Murty, Raymond H. Plaut, Aubrey B. Poore, Calvin J. Ribbens, Yung Shin, and Gopal Vasudevan. This work was supported in part by Department of Energy grant DE-FG05-88ER25068, NASA grant NAG-1-1079, National Science Foundation grant CTS-8913198, and Air Force Office of Scientific Research grant 89-0497.

14. References.

[1] E. Allgower and K. Georg, *Simplicial and continuation methods for approximating fixed points*, SIAM Rev., 22 (1980), pp. 28–85.

[2] D. C. S. Allison, A. Chakraborty, and L. T. Watson, *Granularity issues for solving polynomial systems via globally convergent algorithms on a hypercube*, J. Supercomputing, 3 (1989), pp. 5–20.

[3] D. C. S. Allison, S. Harimoto, and L. T. Watson, *The granularity of parallel homotopy algorithms for polynomial systems of equations*, Internat. J. Comput. Math., 29 (1989), pp. 21–37.

[4] J.-F. M. Barthelemy and M. F. Riley, *Improved multi-level optimization approach for the design of complex engineering systems*, AIAA J., 26 (1988), pp. 353–360.

[5] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[6] S. C. Billups, *An augmented Jacobian matrix algorithm for tracking homotopy zero curves*, M.S. Thesis, Dept. of Computer Sci., VPI & SU, Blacksburg, VA, Sept., 1985.

[7] S. N. Chow, J. Mallet-Paret, and J. A. Yorke, *Finding zeros of maps: Homotopy methods that are constructive with probability one*, Math. Comput., 32 (1978), pp. 887–899.

[8] F. J. Drexler, *Eine Methode zur Berechung sämtlicher Lösunger von Polynomgleichungessystemen*, Numer. Math., 29 (1977), pp. 45–58.

[9] G. H. Ellis and L. T. Watson, *A parallel algorithm for simple roots of polynomials*, Comput. Math. Appl., 10 (1984), pp. 107–121.

[10] W. Fulton, *Intersection Theory*, Springer-Verlag, New York, 1984.

[11] V. Guillemin and A. Pollack, *Differential Topology*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[12] M. M. Kostreva, *Nonconvexity in noncooperative game theory*, Int. J. Game Theory, 18 (1989), pp. 247–259.

[13] M. M. Kostreva and L. A. Kinard, *A differentiable homotopy approach for solving polynomial optimization problems and noncooperative games*, Preprint, Dept. Math. Sciences, Clemson Univ., Clemson, SC, 1987.

[14] G. Kreisselmeier and R. Steinhauser, *Systematic control design by optimizing a vector performance index*, Proc. IFAC Symp. on Computer Aided Design of Control Systems, Zurich, Switzerland, (1979), pp. 113–117.

[15] M. Kubicek, *Dependence of solutions of nonlinear systems on a parameter*, ACM Trans. Math. Software, 2 (1976), pp. 98–107.

[16] T. Y. Li, T. Sauer, and J. A. Yorke, *Numerical solution of a class of deficient polynomial systems*, SIAM J. Numer. Anal., 24 (1987), pp. 435–451.

[17] ——, *Numerically determining solutions of systems of polynomial equations*, Bull. AMS, 18 (1988), pp. 173–177.

[18] O. L. Mangasarian, *Nonlinear Programming*, McGraw-Hill, New York, 1969.

[19] ——, *Equivalence of the complementarity problem to a system of nonlinear equations*, SIAM J. Appl. Math., 31 (1976), pp. 89–92.

[20] R. Mejia, *CONKUB: A conversational path-follower for systems of nonlinear equations*, J. Comput. Phys., 63 (1986), pp. 67–84.

[21] A. P. Morgan, *A method for computing all solutions to systems of polynomial equations*, ACM Trans. Math. Software, 9 (1983), pp. 1–17.

[22] ——, *A transformation to avoid solutions at infinity for polynomial systems*, Appl. Math. Comput., 18 (1986), pp. 77–86.

[23] ——, *A homotopy for solving polynomial systems*, Appl. Math. Comput., 18 (1986), pp. 87–92.

[24] ——, *Solving polynomial systems using continuation for engineering and scientific problems*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[25] A. P. Morgan and A. J. Sommese, *A homotopy for solving general polynomial systems that respects m-homogeneous structures*, Appl. Math. Comput., 24 (1987), pp. 101–113.

[26] ——, *Computing all solutions to polynomial systems using homotopy continuation*, Appl. Math. Comput., 24 (1987), pp. 115–138.

[27] ——, *Coefficient parameter polynomial continuation*, Appl. Math. Comput., 29 (1989), pp. 123–160.

[28] A. P. Morgan, A. J. Sommese, and L. T. Watson, *Finding all isolated solutions to polynomial systems using HOMPACK*, ACM Trans. Math. Software, 15 (1989), pp. 93–122.

[29] A. P. Morgan and L. T. Watson, *A globally convergent parallel algorithm for zeros of polynomial systems*, Nonlinear Anal., 13 (1989), pp. 1339–1350.

[30] W. Pelz and L. T. Watson, *Message length effects for solving polynomial systems on a hypercube*, Parallel Comput., 10 (1989), pp. 161–176.

[31] A. B. Poore and D. Soria, *Continuation algorithms for linear programming*, manuscript.

[32] A. B. POORE AND Q. AL-HASSAN, *The expanded Lagrangian system for constrained optimization problems*, SIAM J. Control Optim., 26 (1988), pp. 417–427.

[33] W. C. RHEINBOLDT AND J. V. BURKARDT, *Algorithm 596: A program for a locally parameterized continuation process*, ACM Trans. Math. Software, 9 (1983), pp. 236-241.

[34] H. SCARF, *The Computation of Economic Equilibria*, Yale University Press, (1973).

[35] L. F. SHAMPINE AND M. K. GORDON, *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, W. H. Freeman, San Francisco, 1975.

[36] Y. S. SHIN, R. T. HAFTKA, L. T. WATSON, AND R. H. PLAUT, *Tracing structural optima as a function of available resources by a homotopy method*, Comput. Methods Appl. Mech. Engrg., 70 (1988), pp. 151–164.

[37] ——, *Design of laminated plates for maximum buckling load*, J. Composite Materials, 23 (1989), pp. 348–369.

[38] G. VASUDEVAN, L. T. WATSON, AND F. H. LUTZE, *A homotopy approach for solving constrained optimization problems*, Tech. Rep. 88–50, Dept. of Computer Sci., VPI&SU, Blacksburg, VA, 1988.

[39] L. T. WATSON, *A globally convergent algorithm for computing fixed points of C^2 maps*, Appl. Math. Comput., 5 (1979), pp. 297–311.

[40] ——, *Solving the nonlinear complementarity problem by a homotopy method*, SIAM J. Control Optim., 17,1 (1979), pp. 36–46.

[41] ——, *Computational experience with the Chow-Yorke algorithm*, Math. Programming, 19 (1980), pp. 92–101.

[42] ——, *An algorithm that is globally convergent with probability one for a class of nonlinear two-point boundary value problems*, SIAM J. Numer. Anal., 16 (1979) 394–401.

[43] ——, *Solving finite difference approximations to nonlinear two-point boundary value problems by a homotopy method*, SIAM J. Sci. Stat. Comput., 1 (1980) 467–480.

[44] ——, *Numerical linear algebra aspects of globally convergent homotopy methods*, Tech. Report TR–85–14, Dept. of Computer Sci., VPI&SU, Blacksburg, VA, 1985, and SIAM Rev., 28 (1986), pp. 529–545.

[45] ——, *Globally convergent homotopy methods: a tutorial*, Appl. Math. Comput., 31BK (1989), pp. 369–396.

[46] L. T. WATSON, S. C. BILLUPS, AND A. P. MORGAN, *HOMPACK: A suite of codes for globally convergent homotopy algorithms*, Tech. Rep. 85–34, Dept of Industrial and Operations Eng., Univ. of Michigan, Ann Arbor, MI, 1985, and ACM Trans. Math. Software, 13 (1987), pp. 281–310.

[47] L. T. WATSON, J. P. BIXLER, AND A. B. POORE, *Continuous homotopies for the linear complementarity problem*, Tech. Report TR–87–38, Dept. of Computer Sci., VPI&SU, Blacksburg, VA, 1987.

[48] L. T. WATSON AND D. FENNER, *Chow-Yorke algorithm for fixed points or zeros of C^2 maps*, ACM Trans. Math. Software, 6 (1980), pp. 252–260.

[49] L. T. WATSON AND R. T. HAFTKA, *Modern homotopy methods in optimization*, Comput. Methods Appl. Mech. Engrg., 74 (1989), pp. 289–305.

[50] L. T. WATSON AND M. R. SCOTT, *Solving spline-collocation approximations to nonlinear two-point boundary-value problems by a homotopy method*, Appl. Math. Comput., 24 (1987) 333–357.

[51] L. T. WATSON AND L. R. SCOTT, *Solving Galerkin approximations to nonlinear two-point boundary value problems by a globally convergent homotopy method*, SIAM J. Sci. Stat. Comput., 8 (1987) 768–789.

[52] L. T. WATSON AND C. Y. WANG, *A homotopy method applied to elastica problems*, Internat. J. Solids Structures, 17 (1981), pp. 29–37.

[53] L. T. WATSON AND W. H. YANG, *Some numerical methods for limit analysis in continuum mechanics*, Computer Methods Appl. Mech. Engrg., 15 (1978), pp. 353–364.

[54] ——, *Optimal design by a homotopy method*, Applicable Anal., 10 (1980), pp. 275–284.