SIAG/OPT Views-and-News

A Forum for the SIAM Activity Group on Optimization

Volume 11 Number 1 April 2000

Contents

Case Studies

Case Studies

Globally Convergent Homotopy Methods

Layne Watson
Departments of Computer Science and Mathematics,
Virginia Polytechnic Institute & State University,
Blacksburg, VA 24061-0106 USA, *E-mail address*:
ltw@vt.edu URL: http://www.cs.vt.edu/~ltw/

1. Introduction

Larry Shampine once said in a talk about ODE algorithms that "the implementation of an algorithm is more important than the algorithm itself." This certainly applies to a class of algorithms variously known as continuation, invariant embedding, incremental loading, homotopy, and path following. The ideas are old, appear in topology, nonlinear real analysis, and algebraic geometry, and except for a few notable applications, have not been considered mainstream numerical analysis. But good ideas tend to be reincarnated, and many of the highly touted concepts in interior point methods are just old homotopy ideas reinvented, reimplemented, and (of course) renamed! For many years homotopy methods for nonlinear equations were deemed a theoretical curiosity rather than a practical approach, and this belief was reinforced by horrible early computer implementations. Shampine was dead right, though, as current implementations of homotopy methods are efficient and definitely competitive with other approaches (e.g., quasi-Newton as in MINPACK).

Indeed, in many application areas (robot motion planning, linkage mechanism design, variable geometry trusses, geometric surface intersection in CAD/-CAM, magnetohydrodynamic fluid flow, heavy elastica, fluid-solid interactions, analog circuit simulation), homotopy methods have become the method of choice. For finding all roots of a polynomial system of equations, presently the only serious, production approach is a homotopy algorithm. Successful, nontrivial applications have occurred in nearly every engineering discipline, in biology, computer science, chemistry, control systems, and statistics (see [3] for a sample of applications). The following sections discuss the supporting theory, application to optimization, and available software.

2. Probability-one Globally Convergent Homotopies

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

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

$$\rho(\lambda, x) = \lambda f(x) + (1 - \lambda) g(x) \qquad (1)$$

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

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

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

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

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

All four shortcomings of continuation and homotopy methods have been overcome by probability-one homotopies, proposed in 1976 by Chow, Mallet-Paret, and Yorke [2]. The supporting theory, based on differential geometry, will be reformulated in less technical jargon here.

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

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

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

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

is also transversal to zero.

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

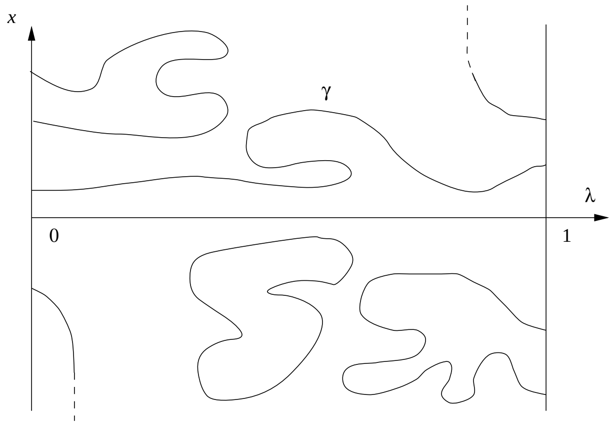

Figure 1. Zero set for $\rho_a(\lambda, x)$ satisfying properties (1)—(4).

Theorem 3 *Let $f : \mathbf{R}^p \to \mathbf{R}^p$ be a C^2 map, $\rho : \mathbf{R}^m \times [0,1) \times \mathbf{R}^p \to \mathbf{R}^p$ a C^2 map, and $\rho_a(\lambda, x) = \rho(a, \lambda, x)$. Suppose that*

(1) ρ is transversal to zero,

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

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

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

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

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

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

Any algorithm for tracking γ from $(0, x_0)$ to $(1, \bar{x})$, based on a homotopy map satisfying the hypotheses of Theorem 3, is called a *globally convergent probability-one homotopy algorithm*. Of course the practical numerical details of tracking γ are nontrivial, and have been the subject of twenty years of research in numerical analysis. Production quality software called HOMPACK90 [4] exists for tracking γ. The distinctions between continuation, homotopy methods, and probability-one homotopy methods are subtle but worth noting. Only the latter are provably globally convergent and (by construction) expressly avoid dealing with singularities numerically, unlike continuation and homotopy methods which must explicitly handle singularities numerically.

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

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

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

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

3. Optimization Homotopies

A few typical convergence theorems for optimization are given next (see the survey in [3] for more examples and references). Consider first the *unconstrained optimization* problem

$$\min_{x} f(x). \tag{3}$$

Theorem 4 *Let $f : \mathbf{R}^n \to \mathbf{R}$ 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 (3).

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

$$\min_{x \ge 0} f(x). \tag{4}$$

This is more general than it might appear because the general convex *quadratic program* reduces to a problem of the form (4).

Theorem 5 *Let $f : \mathbf{R}^n \to \mathbf{R}$ 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 (4).

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

$$x \ge 0, \quad F(x) \ge 0, \quad x^t F(x) = 0. \tag{5}$$

It is interesting that homotopy methods can be adapted to deal with nonlinear inequality constraints and combinatorial conditions as in (5). Define $G : \mathbf{R}^n \to \mathbf{R}^n$ by

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

and let

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

Theorem 6 *Let $F : \mathbf{R}^n \to \mathbf{R}^n$ be a C^2 map, and let the Jacobian matrix $DG(z)$ be nonsingular at every zero of $G(z)$. Suppose there exists $r > 0$ such that $z > 0$ and $z_k = \|z\|_\infty \ge r$ 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 (5).*

Theorem 7 *Let $F : \mathbf{R}^n \to \mathbf{R}^n$ be a C^2 map, and let the Jacobian matrix $DG(z)$ be nonsingular at every zero of $G(z)$. Suppose there exists $r > 0$ such that $z \ge 0$ and $\|z\|_\infty \ge r$ imply $z_k F_k(z) > 0$ for some index k. Then there exists $\delta > 0$ such that for almost all $a \ge 0$ 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 (5).*

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

$$\begin{aligned} &\min && \theta(x) \\ &\text{subject to} && g(x) \le 0, \\ & && h(x) = 0, \end{aligned} \tag{6}$$

where $x \in \mathbf{R}^n$, θ is real valued, g is an m-dimensional vector, and h is a p-dimensional vector. Assume that θ, g, and h are C^2. The *Kuhn-Tucker necessary optimality conditions* for (6) are

$$\begin{aligned} \nabla\theta(x) + \beta^t\nabla h(x) + \mu^t\nabla g(x) &= 0, \\ h(x) &= 0, \\ g(x) &\le 0, \\ \mu &\ge 0, \\ \mu^t g(x) &= 0, \end{aligned} \tag{7}$$

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

$$W(x, \mu) = 0, \tag{8a}$$

where

$$W_i(x, \mu) = -|\mu_i + g_i(x)|^3 + \mu_i^3 - (g_i(x))^3, \tag{8}$$

$$i = 1, \ldots, m. \tag{9}$$

$$\tag{10}$$

Thus the optimality conditions (7) take the form

$$F(x, \beta, \mu) = \begin{pmatrix} [\nabla\theta(x) + \beta^t\nabla h(x) + \mu^t\nabla g(x)]^t \\ h(x) \\ W(x, \mu) \end{pmatrix} = 0. \tag{9}$$

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

$$\rho_a(\lambda, z) = \lambda F(z) + (1 - \lambda)(z - a), \tag{10}$$

where $a \in \mathbf{R}^{n+p+m}$. 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 realistic engineering problems.

Frequently in practice the functions θ, g, and h involve a parameter vector c, and a solution to (6) is known for some $c = c^{(0)}$. Suppose that the problem under consideration has parameter vector $c = c^{(1)}$. Then

$$c = (1 - \lambda)c^{(0)} + \lambda c^{(1)} \tag{11}$$

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

$$\rho_a(\lambda, z) = \lambda F(\lambda, z) + (1 - \lambda)(z - a) \tag{12}$$

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

$$F(\lambda, z) = 0, \tag{13}$$

since the solution $z^{(0)}$ to $F(0, z) = 0$ (the problem corresponding to $c = c^{(0)}$) is known. However, for various technical reasons, (12) is much better than (13).

4. Software

There are several software packages implementing both continuous and *simplicial* homotopy methods; see [1] and [4] for a discussion of some of these packages. A production quality software package written in Fortran 90 is described here. *HOMPACK90* [4] is a Fortran 90 collection of codes for finding zeros or fixed points of nonlinear systems using globally convergent probability-one homotopy algorithms. Three qualitatively different algorithms—ordinary differential equation based, normal flow, quasi-Newton augmented Jacobian matrix—are provided for tracking homotopy zero curves, as well as separate routines for dense and sparse Jacobian matrices. A high level driver for the special case of polynomial systems is also provided. HOMPACK90 features elegant interfaces, use of modules, support for several sparse matrix data structures, and modern

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

Table 1: Taxonomy of homotopy subroutines.

iterative algorithms for large sparse Jacobian matrices.

HOMPACK90 is logically organized in two different ways: by algorithm/problem type and by subroutine level. There are three levels of subroutines. The top level consists of drivers, one for each problem type and algorithm type. The second subroutine level implements the major components of the algorithms such as stepping along the homotopy zero curve, computing tangents, and the end game for the solution at $\lambda = 1$. The third subroutine level handles high level numerical linear algebra such as QR factorization, and includes some LAPACK and BLAS routines. The organization of HOMPACK90 by algorithm/problem type is shown in Table 1, which lists the driver name for each algorithm and problem type.

The naming convention is

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

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

SELECTED_REAL_KIND(13),

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

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

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

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

Many scientific programmers prefer the reverse call paradigm, whereby a subroutine returns to the calling program whenever the subroutine needs certain information (e.g., a function value) or a certain operation performed (e.g., a matrix-vector multiply). Two reverse call subroutines (STEPNX, ROOTNX) are provided for "expert" users. STEPNX is an expert reverse call stepping routine for tracking a homotopy zero curve γ that returns to the caller for all linear algebra, all function and derivative values, and can deal gracefully with situations such as the function being undefined at the requested steplength.

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

REFERENCES

[1] Allgower, E. L., and Georg, K., *Numerical Continuation Methods*, Springer-Verlag, Berlin, 1990.

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

[3] Watson, L. T., *Globally convergent homotopy algorithms for nonlinear systems of equations*, Nonlinear Dynamics, 1,(1990), 143–191.

[4] Watson, L. T., Sosonkina, M., Melville, R. C., Morgan, A. P., and Walker, H. F., *Algorithm 777: HOMPACK90: a suite of Fortran 90 codes for globally convergent homotopy algorithms*, ACM Trans. Math. Software, 23,(1997), 514–549.

Emergency Broadcast Systems Revisited

A.J. Quist
Information Technology and Systems, Delft University of Technology, Delft, The Netherlands (currently working at OM Partners, Brasschaat, Belgium aquist@ompartners.com)
H.N. Post
Future Technology, Baarn, The Netherlands,
Future_Technology@csi.com
M.A. van Hartskamp
Mathematics and Computer Science, Vrije Universiteit Amsterdam, The Netherlands (currently working at Philips Research, Eindhoven, The Netherlands
michael.van.hartskamp@philips.com)

1. Introduction

In the SIAG/OPT Views-and-News Volume 10 Nr. 1, a paper is presented [2] concerning the problem to determine whether a receiving device is within a polygon, the coordinates of which are transmitted as a radio signal to the receiver. If the receiver is within the polygon, it rises a signal, otherwise it keeps silent. In order to determine whether it is in the polygon or not, it is able to use integer arithmetic only. It is assumed that all vertex values, as well as the position of the receiver, lie on an integer grid.

In [2] it is proposed to solve this problem by translating it into a linear program. This program is solved using the revised simplex method, that is refined in a beautiful way as to use only integer arithmetic. It is concluded in [2] that the method works well for small problems, where the polygon has less than 10 vertices, but is impractical for large problems.

At first sight, the article demonstrates a very nice example of the wide applicability and flexibility of mathematical optimization. On the other hand, the problem of determining whether or not a point is in a given polygon is not new. The problem and various extensions of it are well known in for example computational geometry [1]. We encountered the same problem a few years ago in another different context. We were given a database containing all zip code areas of a certain country, described by their vertices. Given this database and a point b, we had to answer the question: 'is b in zip code area g?'. We developed an algorithm based on common sense, that is conceptually simpler than a linear optimization algorithm and computationally much more efficient. Although it is not an optimization algorithm, we like to present it here, in order to prevent the situation in which OR people demonstrate the linear program as an advertisement of their discipline, while the listeners from computer science have a much faster and simpler solution. This holds especially since papers that appeared in SIAG/OPT Views-and-news are especially likely to be used for OR advertisement.

Let us introduce some notation. The polygon is described by the vertices $x^1, ..., x^n$. The location under test is $b = (b_1, b_2)$. Our method requires the additional assumption that the vertices $x^1, ..., x^n$ are given in the right order, i.e. x^i and x^{i+1} are adjacent, and x^n is adjacent to x^1. This additional assumption is necessary since we may deal with arbitrary, not necessarily convex polygons. The method is based on a common sense statement, that for topology specialists may be derived by the Jordan Curve Theorem [3]. It needs a point p that is outside the polygon. Given this point, the number of intersections is counted between the line b-p and the boundary of the polygon. If this number is odd, then b is inside the polygon, otherwise it is outside.

Counting the number of boundary crossings reduces to checking whether line segment $x^i - x^{i+1}$ crosses line segment $b - p$, for $i = 1, ..., n$, where $x^{n+1} = x^1$. A crossing only occurs if x^i and x^{i+1} are on opposite sides of the line through b and p, and

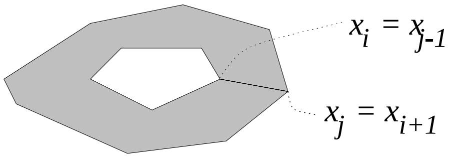

Figure 1: The algorithm also works for this nonconvex polygon, that we actually encountered in our zip code database as the periphere region around a city centre.

simultaneously b and p are on opposite sides of the line through x^i and x^{i+1}. Define the 2×2 matrices A and B_i:

$$A = (b\ p),\ \ B = (x^i\ x^{i+1}),\ \ i = 1, ..., n.$$

Further define $e = (1\ 1)^T$ and

$$\begin{aligned} \alpha_i &= \operatorname{sgn}(e^T \operatorname{adj}(A) x^i - \det(A)) \\ \beta_i &= \operatorname{sgn}(e^T \operatorname{adj}(B_i) b - \det(B_i)) \\ \gamma_i &= \operatorname{sgn}(e^T \operatorname{adj}(B_i) p - \det(B_i)) \end{aligned}$$

Note that α_i is plus or minus 1 depending on whether x^i is on the left or the right side of the line through b and p, or zero if x^i is exactly on the line. Similar interpretations hold for β_i and γ_i. Neglecting the possibility that a point is exactly on the other line, the basic algorithm now reads as follows.

```
1. $c = 0$ {the crossings counter};
2. $p_1 = \max_i(x^i_1) + 1$; $p_2 = \max_i(x^i_2) + 1$;
3. For $i = 1$ to $n$ do
       If $\alpha_i \neq \alpha_{i+1}$ and $\beta_i \neq \gamma_i$ then
           $c = c + 1$;
4. $b$ is inside the polygon if and only if $c$ is odd.
```

The algorithm requires at most $3n$ computations of α's, β's and γ's. This includes the computation of at most $n + 1$ determinants and adjugates of 2×2 matrices, which is perfectly and efficiently possible using integer arithmetic.

The program logic is slightly complicated by considering the case that one or more α's or β's are zero. If for some i $\beta_i = 0$ then b is on the boundary of the polygon. In that case we can immediately stop the algorithm. If one or more $\alpha'_i s$ are zero (i.e., x^i is on the line through b and p) nasty things can happen and a complicated if-structure is needed to cover all cases. But since all points are on an integer grid this case is easily avoided by choosing p as follows:

$$\begin{aligned} p_1 &= b_1 + 1 \\ p_2 &= b_2 + \max_i |b_2 - x^i_2| + 1 \end{aligned}$$

This p is outside the polygon since $p_2 > x^i_2$ for all i, and $\alpha_i = 0$ only if x^i is b. The latter case is already covered by the check for zero $\beta's$. The total algorithm is listed below, where $\alpha's$, $\beta's$ and $\gamma's$ are computed as needed.

```
1. If $\min_i |\beta_i| = 0$ then STOP: $b$ is on the boundary;
2. $c = 0$ {the crossings counter};
3. $p_1 = b_1 + 1$; $p_2 = b_2 + \max_i |b_2 - x^i_2| + 1$;
4. For $i = 1$ to $n$ do
       If $\alpha_i \neq \alpha_{i+1}$ and $\beta_i \neq \gamma_i$ then
           $c = c + 1$;
5. $b$ is inside the polygon if and only if $c$ is odd.
```

The linear time algorithm works also fine for any –possibly nonconvex– polygon, even for a ring like in Figure 1, requires only integer arithmetic, and is very fast.

REFERENCES

[1] Berg, M. de and Kreveld, M. van and M. Overmars and O. Schwarzkopf *Computational Geometry: Algorithms and Applications*, Springer, Berlin, 1997.

[2] M.C. Ferris and T.S. Munson *Linear Programming for Emergency Broadcast Systems*, SIAG/OPT Views-and-News, Vol. 10, No. 1, pp 6–8, 1999.

[3] Ryuji Maehara *The Jordan Curve Theorem Via the Brouwer Fixed Point Theorem* American Mathematical Monthly, Vol. 91, pp. 641–643, 1984.

Author's Response to Emergency Broadcast Systems Revisited

Michael C. Ferris
Computer Sciences Department
University of Wisconsin
1210 West Dayton Street
Madison, Wisconsin 53706

The Quist et al paper uses a line crossing approach to solve the problem. It is interesting in that it treats non-convex regions just as effectively as convex ones. I believe the assumption that the points are given in order could be a drawback in the application, since in general the emergency broadcaster has to generate the points under duress and very quickly. Furthermore, in practice, the operator has to extend the region by adding points to the convex hull as the emergency develops. It is easy for the receivers to deal with extra points in the linear programming approach, whereas a new set of points would have to be generated using the line crossing approach.

Finally, the notion of efficiency. Typically, linear programs require a number of pivots that is some small multiple of the number of rows. In our case, 5 or 6 pivots typically suffice. The computational cost of each pivot is the calculation of a 3 by 3 adjugate, a 3 by 3 determinant and 3n + 39 multiplications. Thus in practice time is negligible for the linear programming approach.

Bulletin

Comments from the Chair and Editor

Thomas F. Coleman, SIAG/OPT Chair
Computer Science
Cornell University
Ithaca, NY 14853
coleman@tc.cornell.edu

Juan C. Meza, Editor
Sandia National Laboratories
P.O. Box 969, MS 9217
Livermore, CA 94551
meza@ca.sandia.gov