Solution of polynomial systems derived from differential equations

E. L. Allgower, Fort Collins, **D. J. Bates**, Notre Dame,
A. J. Sommese, Notre Dame, and **C. W. Wampler**, Warren

May 4, 2005

Abstract

Nonlinear two-point boundary value problems arise in numerous areas of application. The existence and number of solutions for various cases has been studied from a theoretical standpoint. These results generally rely upon growth conditions of the nonlinearity. However, in general, one cannot forecast how many solutions a boundary value problem may possess or even determine the existence of a solution. In recent years numerical continuation methods have been developed which permit the numerical approximation of all complex solutions of systems of polynomial equations. In this paper, numerical continuation methods are adapted to numerically calculate the solutions of finite difference discretizations of nonlinear two-point boundary value problems. The approach taken here is to perform a homotopy deformation to successively refine discretizations. In this way additional new solutions on finer meshes are obtained from solutions on coarser meshes. The complicating issue which the complex polynomial system setting introduces is that the number of solutions grows with the number of mesh points of the discretization. To counter this, the use of filters to limit the number of paths to be followed at each stage is considered.

AMS Subject Classification: 65L10, 65H10, 68W30, 14Q99

Key words: differential equations, boundary value problems, numerical algebraic geometry, homotopy continuation, polynomial systems

1 Introduction

Consider a two-point boundary value problem on the interval $[a, b] \subset \mathbb{R}$,

$$y'' = f(x, y, y'), \tag{1}$$

with boundary conditions $y(a) = \alpha$ and $y(b) = \beta$. The standard central difference approximation with a uniform mesh may be used to approximate solutions to (1). In particular, let N be a positive integer, $h := \frac{b-a}{N+1}$, and $x_i := a + ih$ for $i = 0, \dots, N+1$. Setting $y_0 = \alpha$ and $y_{N+1} = \beta$, the discretization of (1) takes the form of the system $\mathcal{D}_N$:

$$\begin{array}{ccccccl} y_0 & - & 2y_1 & + & y_2 & = & h^2 f(x_1, y_1, \frac{y_2 - y_0}{2h}) \\ \vdots & & \vdots & & \vdots & = & \vdots \\ y_{N-1} & - & 2y_N & + & y_{N+1} & = & h^2 f(x_N, y_N, \frac{y_{N+1} - y_{N-1}}{2h}) \end{array}$$

A solution $y(x)$ of (1) may then be approximated by an N-tuple of real numbers $(y_1, \dots, y_N)$ such that $y_i \approx y(x_i)$ for $i = 1, \dots, N$.

Depending upon the nonlinearity f, equation (1) may have no solutions, a unique solution, multiple solutions, or even infinitely many solutions. There are many existence theorems for solutions of such equations subject to growth conditions on f, but even when existence is known, the number of solutions often is not. Furthermore, a discretization such as $\mathcal{D}_N$ may have spurious solutions that do not converge to a solution to (1) as $N \to \infty$. On the other hand, if f is sufficiently smooth, a solution y to (1) is eventually approximated with $\mathcal{O}(h^2)$ accuracy on the mesh by some solution $\bar{y} \in \mathbb{R}^N$.

The purpose of the present paper is to give a relatively secure numerical technique for finding the solutions of a general class of two-point boundary value problems without requiring highly refined meshes. The technique involves performing successive homotopy deformations between discretizations with increasingly many mesh points, as suggested in [2]. By restricting our attention to problems having polynomial nonlinearity, including the case of a polynomial approximation to a smooth nonlinearity, we can often assure that all solutions are found at each stage of the algorithm. Even when we do not guarantee all solutions, our method generates multiple solutions that in test cases include approximations to all known solutions. While Gröbner basis methods (see [4]) or cellular exclusion methods (see [6]) could be applied to solve the polynomial discretizations, we chose to use homotopy continuation due to its ability to handle polynomial systems in many variables and the ease with which it allows us to generate solutions on a refined mesh from the solutions on the previous mesh. Although other numerical methods treating two-point boundary value problems have been developed (see [7] and [10]), such methods require satisfactory initial solution estimates. The present technique provides such initial estimates.

Here is a sketch of our bootstrapping process, which will be discussed in more detail in the subsequent section:

1. Find all solutions of the discretization $\mathcal{D}_N$ for some small N. The size of N needs only to be large enough that the discretization is consistent; it could be as small as $N = 1$.

2. Discard all unreasonable solutions, e.g., solutions which do not possess properties which exact solutions may be known to have. Let us denote the set of solutions which are kept by $\mathcal{V}_N$.

3. If the mesh size is not yet sufficiently small or the cardinality of $\mathcal{V}_N$ has not yet stabilized, add a mesh point to obtain the discretization $\mathcal{D}_{N+1}$. Use the solutions in $\mathcal{V}_N$ to generate solutions $\mathcal{V}_{N+1}$ of $\mathcal{D}_{N+1}$ and then return to Step 2.

4. Once the mesh size h is sufficiently small and the cardinality of $\mathcal{V}_N$ becomes stable, refine the solutions with a fast nonlinear solver, using starting values obtained by interpolating the solutions in $\mathcal{V}_N$.

This paper focuses primarily on the implementation of Step 3 of the above scheme. In particular, we consider the homotopy function

$$H_{N+1}(y_1, \ldots, y_{N+1}, t) :=$$
$$\begin{bmatrix} y_0 - 2y_1 + y_2 & - & h(t)^2 f\left(x_1(t), y_1, \frac{y_2 - y_0}{2h(t)}\right) \\ & \vdots & \\ y_{N-2} - 2y_{N-1} + y_N & - & h(t)^2 f\left(x_{N-1}(t), y_{N-1}, \frac{y_N - y_{N-2}}{2h(t)}\right) \\ y_{N-1} - 2y_N + Y_{N+1}(t) & - & h(t)^2 f\left(x_N(t), y_N, \frac{Y_{N+1}(t) - y_{N-1}}{2h(t)}\right) \\ y_N - 2y_{N+1} + Y_{N+2}(t) & - & h(t)^2 f\left(x_{N+1}(t), y_{N+1}, \frac{Y_{N+2}(t) - y_N}{2h(t)}\right) \end{bmatrix}$$

with

$$\begin{aligned} y_0 &:= \alpha \\ h(t) &:= t\left(\tfrac{b-a}{N+1}\right) + (1-t)\left(\tfrac{b-a}{N+2}\right) \\ Y_{N+1}(t) &:= (1-t)y_{N+1} + \beta t \\ Y_{N+2}(t) &:= \beta(1-t) \\ x_i(t) &:= a + ih(t), \qquad i = 1, \ldots, N+1. \end{aligned}$$

At $t = 0$ this is the system $\mathcal{D}_{N+1}$. At $t = 1$, it can be interpreted as the system $\mathcal{D}_N$ with a new mesh point having the value y_{N+1} at $x = b$ and a new right-hand boundary at $x = b + h(1)$ having value $Y_{N+2}(1)$. The incompatibility of the old boundary condition at $x = b$ and the new one at $x = b + h(1)$ is accommodated by the presence of both y_{N+1} and Y_{N+1}, which are not necessarily equal. As t goes from 1 to 0, the mesh points are squeezed back inside the interval $[a, b]$, and the right-hand boundary condition $y(b) = \beta$ is transferred from Y_{N+1} to Y_{N+2} as Y_{N+1} is forced to equal y_{N+1}.

To find solutions of $\mathcal{D}_{N+1}$, we use continuation to track solutions of H_{N+1} as t goes from 1 to 0. At $t = 1$, we have a list $\mathcal{V}_N$ of solutions $(y_1, \ldots, y_N)$ satisfying the first N equations of H_{N+1}, while the final equation is

$$y_N - 2y_{N+1} = h(t)^2 f\left(b, y_{N+1}, \frac{-y_N}{2h}\right),$$

which is the only place where y_{N+1} appears. For each solution of $(y_1, \ldots, y_N)$ in $\mathcal{V}_N$, we may use this equation to find corresponding solution values for y_{N+1}. These are the start points of continuation paths leading to solutions of $\mathcal{D}_{N+1}$.

The framework above does not change in any of its essentials if we prescribe a different function for $Y_{N+2}(t)$. For example, the constant function $Y_{N+2}(t) := \beta$ was used for all examples below. Although other alternatives are theoretically feasible, none were tested. The essential feature of $Y_{N+2}(t)$ is that it goes to β as t goes to 0.

By the implicit function theorem, a nonsingular solution $y = y^*$ to $H_{N+1}(y, 1) = 0$ will continue uniquely in the neighborhood of $t = 1$ to a nonsingular solution path $y(t)$ satisfying $H_{N+1}(y(t), t) = 0$ with $y(1) = y^*$. This does not mean, however, that the path remains nonsingular all the way to $t = 0$, which is what we require to follow the path reliably with numerical continuation. To skirt this difficulty, as discussed in Chapter 7 of [11], it is sufficient to insert a random $\gamma \in \mathbb{C}$ into the homotopy to obtain the variant

$$H_{N+1}(y_1, \ldots, y_{N+1}, t) := \left[\begin{array}{rcl} \Gamma(t)\,(y_0 - 2y_1 + y_2) & - & h(t)^2 f\left(x_1(t), y_1, \frac{y_2 - y_0}{2h(t)}\right) \\ & \vdots & \\ \Gamma(t)\,(y_{N-2} - 2y_{N-1} + y_N) & - & h(t)^2 f\left(x_{N-1}(t), y_{N-1}, \frac{y_N - y_{N-2}}{2h(t)}\right) \\ \Gamma(t)\,(y_{N-1} - 2y_N) + Y_{N+1}(t) & - & h(t)^2 f\left(x_N(t), y_N, \frac{Y_{N+1}(t) - y_{N-1}}{2h(t)}\right) \\ \Gamma(t)\,(y_N - 2y_{N+1} + \beta) & - & h(t)^2 f\left(x_{N+1}(t), y_{N+1}, \frac{\beta - y_N}{2h(t)}\right) \end{array} \right] \tag{2}$$

with

$$\begin{array}{rcl} \Gamma(t) & := & \gamma^2 t + (1 - t) \\ h(t) & := & \gamma t \left(\frac{b-a}{N+1}\right) + (1 - t)\left(\frac{b-a}{N+2}\right) \\ Y_{N+1}(t) & := & (1 - t) y_{N+1} + \gamma^2 \beta t \\ x_i(t) & := & a + i h(t), \qquad i = 1, \ldots, N + 1. \end{array}$$

The work of the second author was supported by the National Science Foundation under Grant No. 0105653 and Grant No. 0410047; and a fellowship from the Arthur J. Schmitt Foundation. The work of the third author was supported by the National Science Foundation under Grant No. 0105653 and Grant No. 0410047; and the Duncan Chair of the University of Notre Dame. The work of the fourth author was supported by the National Science Foundation under Grant No. 0410047.

2 The case of polynomial nonlinearity

Let's specialize the homotopy of equation (2) to the case when $f(x, y, y')$ is a real polynomial $p(y)$. Then, the right-most term of the i^{th} entry in $H_{N+1}(y, t)$ becomes just $h^2(t)p(y_i)$. This restriction to the polynomial case allows us to conveniently obtain the start points for $H_{N+1}(y_1, \ldots, y_{N+1}, 1) = 0$ by solving the polynomial

$$y_N - 2y_{N+1} + \beta - \left(\frac{b-a}{N+1}\right)^2 p(y_{N+1}) = 0$$

for y_{N+1} given y_N from the solutions in $\mathcal{V}_N$.

Let $d = \deg p(y)$. We see that, in general, over the complex numbers, we will obtain d values of y_{N+1} for every point in $\mathcal{V}_N$. Suppose that at each stage of the algorithm these all continue to finite, nonsingular solutions of $\mathcal{D}_{N+1}$. Then, the solution list $\mathcal{V}_N$ will have d^N entries. While this gives an exhaustive enumeration of the solutions of the discretized problem, the exponential growth in the length of the solution list cannot be practically sustained as N increases. However, it is often the case that most of the solutions at a given stage do not exhibit various properties required of solutions to the two-point boundary value problem at hand, leading to filtering rules. Depending upon the problem at hand, there are a variety of filtering rules that may be implemented to determine which solutions in $\mathcal{V}_N$ may be discarded as start solutions for the subsequent homotopy.

For small N we can contemplate retaining all solutions. It is reasonable to ask whether the above procedure is guaranteed to generate all solutions of the discretized system. The answer, in general, is no, but we can say that if $\mathcal{V}_N$ has d^N distinct, nonsingular solutions, then it is clear that all solutions have been found, as Bézout's theorem states that this is the greatest number possible. Indeed, for our test problems, we have found that this behavior is typical.

For larger N, a filter becomes necessary. One that is always available is to take the discretization of the derivative of $y''' = p'(y)y'$ of $y'' = p(y)$, and throw away $y \in \mathcal{V}_N$ for which this is large. To get the discretization we could use the central difference approximations

$$y'(x_i) = \frac{y_{i+1} - y_{i-1}}{2h},$$

and

$$y'''(x_i) = \frac{y_{i+2} - 2y_{i+1} + 2y_{i-1} - y_{i-2}}{2h^3}$$

applied only at the mesh points $y_2, \ldots, y_{N-1}$. So we would throw away the point $z = (y_1, \ldots, y_N) \in \mathcal{V}_N$ if

$$\sum_{i=2}^{N-1} \left| \frac{y_{i+2} - 2y_{i+1} + 2y_{i-1} - y_{i-2}}{2h^3} - p'(y_i)\frac{y_{i+1} - y_{i-1}}{2h} \right| > \epsilon_2$$

for some $\epsilon_2 > 0$. Naturally, one drawback to such a filter is the need to specify ϵ_2.

Other filters may be derived from known properties of the solutions of the problem at hand. For example, it may be known that solutions are symmetric about $x = \frac{a+b}{2}$, are always positive, oscillate with a specific period, or exhibit some other easily-detected behavior. For example, a filter based on symmetry is considered in Section 3.2. Although one may be tempted to discard solutions having nonzero complex part, this is not a valid filtering rule. The problem in Section 3.3 below has non-real solutions in $\mathcal{V}_N$ that are tracked to real solutions in $\mathcal{V}_{N+1}$. Similarly, it is possible that oscillating solutions may arise from a sequence of non-oscillating solutions and that similar problems may occur with other filters. Thus, the use of filters may be computationally beneficial, but with it comes the risk of not finding all real solutions to the problem.

Thus we have the final version of the algorithm:

Algorithm 1

1. For $N = 1$, $H_1(y_1, 1)$ is a single polynomial in y_1, which may be solved with any one-variable method to produce $\mathcal{V}_1$.
2. For $N = 2, 3, ...$, until some desired behavior has occurred:
   (a) Form the homotopy $H_N(y_1, \ldots, y_N, t)$.
   (b) Solve the last polynomial of $H_N(y_1, \ldots, y_N, t)$ for y_N using each solution in $\mathcal{V}_{N-1}$, thereby forming the set S of the start solutions for $H_N(y_1, \ldots, y_N, t)$.
   (c) Track all paths beginning at points in S at $t = 1$. The set of endpoints of these paths is $\mathcal{V}_N$.
   (d) If desired, apply a filter to $\mathcal{V}_N$ to reduce the number of paths to be tracked in stage $N + 1$.
3. Refine the solutions with a nonlinear solver, if desired.

3 Numerical experiments

The following experiments were run using Bertini, a software package under development by the last three authors for the study of numerical algebraic geometry. Although Bertini was written to make use of multiprecision adaptively, each of the following experiments ran successfully using only 16 digits of precision.

In the following, N denotes the number of mesh points, SOLS(N) denotes the total number of solutions (real or complex), and REAL(N) denotes the number of real solutions. For $N > 1$, the number of paths tracked from stage $N-1$ is $d \cdot \mathrm{SOLS}(N-1)$. A solution is considered to be real if the imaginary part at each mesh point is zero to at least eight digits.

3.1 A basic example

As a first example, consider the following two-point boundary value problem

$$y'' = 2y^3 \tag{3}$$

with boundary conditions $y(0) = \frac{1}{2}$ and $y(1) = \frac{1}{3}$.

There is a unique solution, $y = \frac{1}{x+2}$, to (3). Our method produces one real solution among a total of 3^N solutions found for $N = 1, \ldots, 9$. Furthermore, the error between the computed solution and the unique exact solution behaves as $\mathcal{O}(h^2)$. Refer to Table 1 for details.

N	Maximal error at any mesh point	h^2	Maximal error/h^2
3	1.570846e-04	4.000000e-02	3.927115e-03
4	1.042635e-04	2.777778e-02	3.753486e-03
5	7.069710e-05	2.040816e-02	3.464158e-03
6	5.348790e-05	1.562500e-02	3.423226e-03
7	4.078910e-05	1.234568e-02	3.303917e-03
8	3.230130e-05	1.000000e-02	3.230130e-03
9	2.624560e-05	8.264463e-03	3.175718e-03

Table 1: Evidence of $\mathcal{O}(h^2)$ convergence for Problem (3).

3.2 A more sophisticated example

Consider the problem

$$y'' = -\lambda \left(1 + y^2\right) \tag{4}$$

with zero boundary conditions, $y(0) = 0$ and $y(1) = 0$, and $\lambda > 0$.

According to [8], any solutions to this problem must be symmetric about $x = \frac{1}{2}$, so we have a special filter. Furthermore, it is known that there are two solutions if $\lambda < 4$, a unique solution if $\lambda = 4$, and no solutions if $\lambda > 4$. Without using a filter, the expected number of real solutions in the first and last cases were confirmed computationally (for $\lambda = 2$ and $\lambda = 6$ with $N \leq 17$), and the computed solutions were symmetric as anticipated. From Bézout's theorem, one would expect to find at most 2^N complex solutions at each stage N, and this is precisely the total number of complex solutions found. When $\lambda \approx 4$, the Jacobian of the associated polynomial system is rank-deficient, so regular path-tracking techniques fail.

Tracking all 2^{17} paths for $N = 17$ took just under an hour of CPU time on a single processor Pentium 4, 3 GHz machine running Linux. At this

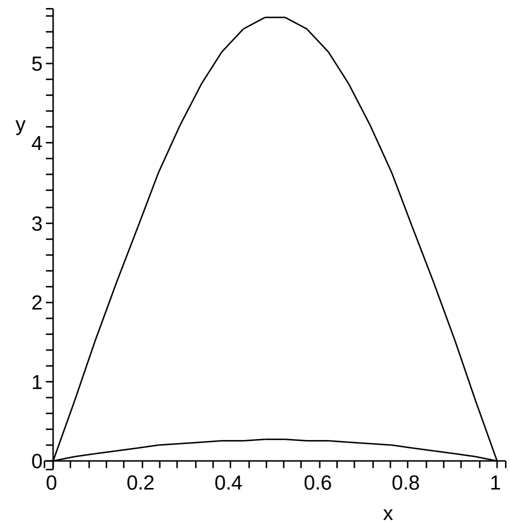

Figure 1: The real solutions of (4) with $N = 20$.

rate, ignoring the time-consuming data management part of the algorithm, it would take well over one year to track all 2^{30} paths for $N = 30$ mesh points. As discussed in Section 2, filtering rules may be used to dramatically reduce the number of paths to be tracked at each stage. A filter forcing $||y_1| - |y_N|| < 10^{-8}$ was applied to the case $\lambda = 2$. This cut the path-tracking time to less than half a second for $N = 17$ mesh points. This drastic reduction in time for path-tracking as well as data management allowed for the confirmation of the existence of two real solutions for up to 100 mesh points. Despite the size of the polynomial system when $N = 100$, each path took less than 4 seconds to track from $t = 1$ to $t = 0$. A graph of the two real solutions for $N = 20$ mesh points is given in Figure 1.

3.3 A problem with infinitely many solutions

It was shown in [3] that the two-point boundary value problem

$$y'' = -\lambda y^3, \qquad y(0) = y(1) = 0, \tag{5}$$

with $\lambda > 0$ has infinitely many oscillating real solutions on the interval $[0, 1]$. Moreover, the solutions occur in pairs in the sense that $-y$ is a solution whenever y is a solution. Hence, together with the trivial solution $y = 0$, we expect always to have an odd number of solutions. That was confirmed computationally, as shown in Table 2. Only the case of $\lambda = 1$ is displayed as all other cases are identical modulo scaling. It may be observed that the number of real solutions found by Bertini grows without bound for this problem, as the number of mesh points increases. In fact, beyond some small value of N, the number of real solutions approximately doubles for each subsequent value of N.

N	SOLS(N)	REAL(N)
1	3	3
2	3	3
3	9	3
4	27	7
5	81	11
6	243	23
7	729	47
8	2187	91

Table 2: Solutions of (5)

3.4 The Duffing problem

One representation (see [5]) of the Duffing problem is the two-point boundary value problem

$$y'' = -\lambda \sin(y) \tag{6}$$

on the interval $[0, 1]$ with $y(0) = 0$, $y(1) = 0$, and $\lambda > 0$. Since our attention is restricted to polynomial nonlinearity only, we approximate $\sin(y)$ by truncating its power series expansion, yielding the problem

$$y'' = -\lambda \left(y - \frac{y^3}{6} \right) \tag{7}$$

using two terms or

$$y'' = -\lambda \left(y - \frac{y^3}{6} + \frac{y^5}{120} \right) \tag{8}$$

using three terms.

It is known that there are $2k + 1$ real solutions to the exact Duffing problem (6) when $k\pi < \lambda < (k+1)\pi$. For a given value of λ, the $2k + 1$ real solutions include the trivial solution $y \equiv 0$ and k pairs of solutions $(y_1(x), y_2(x))$ such that $y_1(x) = -y_2(x)$. Each pair oscillates with a different period. As two- and three-term Taylor series truncations for $\sin(y)$ do not approximate $\sin(y)$ well outside of a small neighborhood, the solutions to (7) and (8) may behave quite differently than those of (6).

Table 3 indicates the number of real solutions found for problems (7) and (8) for $\lambda = 0.5\pi$, 1.5π, and 2.5π. All solutions have either odd or even symmetry about $x = \frac{1}{2}$, so we again used the filter $||y_1| - |y_N|| < 10^{-8}$. The filter was first applied when $N = 4$, so the number of real solutions reported in each case of Table 3 is the number of real solutions found for $N \geq 5$. For

$\lambda = 0.5\pi$ and $\lambda = 1.5\pi$, there were more real solutions found for (8) than predicted for the exact problem (6). However, the computed solutions in each case included one pair of solutions that oscillated wildly. These poorly-behaved solutions are readily identified by the y''' filter discussed in Section 2: for $N = 25$ mesh points, they had residuals four orders of magnitude larger than those of the well-behaved solutions.

λ	$f(y) = y - \frac{y^3}{6}$	$f(y) = y - \frac{y^3}{6} + \frac{y^5}{120}$	$f(y) = \sin(y)$
0.5π	1	3	1
1.5π	1	5	3
2.5π	1	5	5

Table 3: Number of real solutions for approximations of the Duffing problem.

3.5 The Bratu problem

The Bratu problem on the interval $[0, 1]$ has the form

$$y'' = -\lambda e^y, \qquad y(0) = y(1) = 0, \tag{9}$$

with $\lambda > 0$. As in the case of the Duffing problem, we make the right-hand side polynomial by truncating the power series expansion of e^y, yielding

$$y'' = -\lambda \left(1 + y + \frac{y^2}{2}\right) \tag{10}$$

As discussed in [5], there are two real solutions if λ is near zero and no real solutions if λ is large. The real solutions for small λ are symmetric and nonnegative. The expected number and properties of the real solutions in the cases of $\lambda = 0.5$ and $\lambda = 10$ were confirmed, and, as anticipated, 2^N total solutions were found in each case for $N = 1, \ldots, 15$.

4 Discussion

A new algorithm for finding the real solutions of a two-point boundary value problem has been presented, and several examples have been documented under the assumption of polynomial nonlinearity. Furthermore, the use of filtering rules to drastically reduce the computational work has been considered. In each example presented, the number of real solutions predicted by theory has been confirmed computationally, although it was seen that the use of filters may effect the number of real solutions discovered.

There are several variations to the algorithm that could be considered in the future. A more detailed analysis of the benefits and drawbacks of the use of filters could be made. Also, it is possible to add extra mesh points at the left-hand end or middle of the interval rather than the right. Similarly, one new mesh point could be added to each end simultaneously, yielding $(\deg p(y))^2$ starting solutions for each solution from the previous stage. For that matter, non-uniform grids could be analyzed with only mild changes to the formulation. A similar algorithm could also be developed for systems of differential equations.

References

[1] E. L. Allgower. (1975). On a discretization of $y'' + \lambda y^k = 0$. In *Topics in numerical analysis,* II *(Proc. Roy. Irish Acad. Conf., Univ. College, Dublin, 1974)* (pp. 1–15). London: Academic Press.

[2] E. L. Allgower. (1981). A survey of homotopy methods for smooth mappings. In *Numerical solution of nonlinear equations (Bremen, 1980)*, Vol. 878 of *Lecture Notes in Math.* (pp. 1–29). Berlin: Springer.

[3] L. Collatz. (1966). *Differentialgleichungen. Eine Einführung unter besonderer Berücksichtigung der Anwendungen*, Dritte überarbeitete und erweiterte Auflage. Leitfaden der angewandten Mathematik und Mechanik, Band 1. Stuttgart: B. G. Teubner.

[4] D. Cox, J. Little and D. O'Shea. (1998). *Using algebraic geometry*, Vol. 185 of *Graduate Texts in Mathematics*. New York: Springer-Verlag.

[5] H. T. Davis. (1962). *Introduction to nonlinear differential and integral equations*. New York: Dover.

[6] K. Georg. (2001). Improving the efficiency of exclusion algorithms. *Adv. Geom., 1*(2), 193–210.

[7] H. B. Keller. (1992). *Numerical methods for two-point boundary value problems*. New York: Dover. Corrected reprint of the 1968 edition.

[8] T. Laetsch. (1971). On the number of solutions of boundary value problems with convex nonlinearities. *J. Math. Anal. Appl., 35*, 389–404.

[9] T. Y. Li. (2003). Numerical solution of polynomial systems by homotopy continuation methods. In Vol. 11 of *Handbook of numerical analysis* (pp. 209–304). Amsterdam: North-Holland.

[10] S. M. Roberts and J. S. Shipman. (1972). *Two-point boundary value problems: shooting methods*. New York: Elsevier.

[11] A. J. Sommese and C. W. Wampler. (2005). *Numerical solution of systems of polynomials arising in engineering and science*. Singapore: World Scientific.

[12] J. Stoer and R. Bulirsch. (2002). *Introduction to numerical analysis.* Vol. 12 of *Texts in Applied Mathematics.* New York: Springer-Verlag, third edition. Translated from the German by R. Bartels, W. Gautschi and C. Witzgall.

Eugene L. Allgower
Department of Mathematics
Colorado Sate University
Fort Collins, CO 80523-1874
USA
allgower@math.colostate.edu

Daniel J. Bates
Department of Mathematics
University of Notre Dame
Notre Dame, IN 46556-4618
USA
dbates1@nd.edu

Andrew J. Sommese
Department of Mathematics
University of Notre Dame
Notre Dame, IN 46556-4618
USA
sommese@nd.edu

Charles W. Wampler
General Motors Research and Development
Mail Code 480-106-359
30500 Mound Road
Warren, MI 48090-9055
USA
Charles.W.Wampler@gm.com