Numer. Math. 63, 391–409 (1992)

Numerische Mathematik

A power series method for computing singular solutions to nonlinear analytic systems

Alexander P. Morgan[1], Andrew J. Sommese[2], and Charles W. Wampler[1]

[1] Mathematics Department, General Motors Research Laboratories, Warren, MI 48090, USA
[2] Mathematics Department, University of Notre Dame, Notre Dame, IN 46556, USA

Received September 20, 1991

Summary. Given a system of analytic equations having a singular solution, we show how to develop a power series representation for the solution. This series is computable, and when the multiplicity of the solution is small, highly accurate estimates of the solution can be generated for a moderate computational cost. In this paper, a theorem is proven (using results from several complex variables) which establishes the basis for the approach. Then a specific numerical method is developed, and data from numerical experiments are given.

Mathematics Subject Classification (1991): 65H10

1. Introduction

Let $f(z) = 0$ denote a system of n analytic equations in n unknowns with a (possibly singular) solution z^*. In this paper we develop a method to generate an accurate approximation to z^*.

In [16] we showed that z^* can be expressed as an integral of a nonsingular curve. This integral method works well but is relatively expensive to compute. In this paper we generalize the approach of [16], presenting a more fundamental power series representation. This results in two advances: First, the integral method is simplified (in practice) via the circular sample (Sect. 3.3 below). Second, a new method distinct from the integral method can now be developed. The new method is less expensive to implement and often just as effective. Thus, by generalizing the result of [16] we obtain a new flexibility. The underlying theoretical structure from several complex variables is the same as before, but its expression in Theorem 1 here is clarifying and useful numerically.

The theorem which establishes the new framework is stated in Sect. 2 and proven in the Appendix. The resulting numerical methods exploit ideas from homotopy con-

Correspondence to: A.J. Sommese

tinuation [1, 6, 10, 20, 25, 26] and standard techniques for fitting polynomials to data. This is described in Sect. 3. Section 4 gives a particular implementation and the details of numerical experiments in solving two simple test problems. The results of solving two difficult polynomial systems from the kinematics of mechanisms are presented in Sect. 5. Our summary and conclusions are in the final section.

The context of this paper is essentially the same as [16], which is recommended for expository and background material. The paper [17] puts forward a different approach (with a similar theoretical basis) in the more restricted context of *polynomial* rather than *analytic* systems.

Although the theory is cast in a *complex* analytic context, the resulting numerical method can be developed in an entirely *real* analytic framework, if the problem being solved is real. Consequently, homotopy and path tracking software already developed in real arithmetic need not be converted to complex. See, for example, the second test problem in Sect. 4.

For background on other methods of computing singular solutions, see [2–5, 7, 8, 18, 19], most of which address Newton's method variants which can be proven to converge under restrictions on the rank of the Jacobian matrix of the system at the solution (e.g., co-rank 1).

2. Theory

See [16] for the definitions of a *geometrically isolated* solution and the *multiplicity* of a solution. Note that the multiplicity of a solution is defined only if it is geometrically isolated.

As in [16], let $h(z,t)$ be a system of complex analytic functions, with $z \in D$ where D is an open set in C^n and $t \in D_0 \subseteq C^1$, where D_0 is an open set in C^1 containing $[0,1]$, and $h(z,t) \in C^n$ for all $(z,t) \in D \times D_0$. Assume[1]

1. $h(z,0) \equiv f(z)$ for all $z \in D$,
2. $z^*, z^\sigma \in D$ with $h(z^*,0) = 0$ and $h(z^\sigma,1) = 0$,
3. $h^{-1}(0)$ contains a connected complex curve $K \subseteq D \times D_0$ containing z^σ and z^*, so that there is a smooth path $z(t)$ with $t \in [0,1]$ and $z([0,1]) \subseteq K$ such that $z(1) = z^\sigma$, $z(0) = z^*$, and $dh(z(t),t)$ has rank n for $t \in (0,1]$.

Here $dh(z,t)$ denotes the $n \times (n+1)$ complex matrix of partial derivatives of h with respect to $z_1, \ldots, z_n$ and t. Note that $h(z(t),t) = 0$ for $t \in [0,1]$. It is routine to generate such an h, given f, z^σ, and z^*, if z^* is a geometrically isolated solution to $f(z) = 0$ and z^σ is close enough to z^*. See [16], Sect. 3.

We now have our main theorem:

Theorem 1. *There is a $\delta > 0$, a smallest positive integer c, and a power series*

$$\mathcal{Z}(s) = \sum_{k=0}^{\infty} a_k s^k \tag{1}$$

convergent for $|s| < \delta$ such that

$$\mathcal{Z}(s) = z(s^c) \tag{2}$$

[1] We have made one change of notation from [16]: "t" in this paper is "1-t" in [16].

for $s \in [0, \delta]$.

The proof is given in the Appendix.

Remarks.

1. It follows that $a_0 = z(0) = \mathcal{Z}(0) = z^*$.
2. Informally, Theorem 1 says that $z(t)$ is "analytic in the c^{th} root of t." Generally, $z(t)$ will not be "analytic in t."
3. The c is the *cycle number* of the path $z(t)$ as defined in [16]. If z^* is geometrically isolated, then z^* has a multiplicity, m, and $c \leq m$.

Here is a simple example that illustrates the theorem: Let $f(z) = z^m$, with singular solution $z^* = 0$ of multiplicity m. Let

$$h(z, t) = t(z^m - 1) + (1 - t)z^m.$$

Then

$$z(t) = u \sqrt[m]{t},$$

where u is a primitive m^{th} root of unity. Here, $c = m$, $a_0 = 0$, $a_1 = u$, and $s^m = t$.

Note that $z(s^c)$ is defined only for real s, but $\mathcal{Z}(s)$ is defined and holomorphic for all complex s. By the theorem, $z(s^c) = \mathcal{Z}(s)$ on an interval, so $h(\mathcal{Z}(s), s^c) = 0$ on that same interval. Thus, by the analyticity of h and $\mathcal{Z}$,

$$h(\mathcal{Z}(s), s^c) = 0 \tag{3}$$

for all complex s with $|s| < \delta$. We have $z^* = \mathcal{Z}(0)$. The essence of our numerical method will be to sample $\mathcal{Z}(s)$ in a (possibly but not necessarily complex) neighborhood of $s = 0$, fit a power series to the sampled points, and evaluate the series at $s = 0$. We will be able to carry this out by applying standard numerical prediction-correction methods to Eq. (3).

3. The numerical method: creating an adaptive end game

3.1 Overview

In the numerical method we are considering, a homotopy is chosen that obeys the three conditions given at the beginning of Sect. 2 (see [16], pp. 673–674]). Then the path converging to z^* is tracked from $t = 1$ to $t = \tau$, where τ is close to 0 but not too close. Finally, an *adaptive end game*, constructed using the framework established by Theorem 1, completes the path tracking and gives a numerical estimate of z^*.

Tracking paths (solutions to $h(z, t) = 0$) is a standard numerical procedure. Using the Jacobian $dh(z, t)$, we predict ahead along the tangent to the path and then correct back to the path using Newton's method. An adaptive step size in the predictor provides stability and efficiency. Without an adaptive end game, the selection of τ would be critical to success. However, with the adaptive end game in place, we need only pick a conservatively large τ to initiate it. (Typically, we might take the τ to be 10^{-2} or 10^{-3}.)

The end game samples z for $t \in [0, \tau]$ and uses this sample to estimate c, then samples $\mathcal{Z}$ to obtain the best estimate of z^*. The following tasks must be accomplished:

A : Find the end game operating range.
B : Estimate c.
C : Estimate z^*.
D : Evaluate the quality of the estimate of z^*.

The end game operating range is defined below. There are several strategies for accomplishing these tasks, and we shall discuss in general terms what is involved in this subsection. Then, in Sects. 3.2–3.5, several topics of special interest and utility are presented in more detail.

In developing numerical methods based on Theorem 1, it is helpful to have the concept of the *end game operating range*. A complex number s is in the end game operating range if $|s|$ is small enough so that Theorem 1 holds (i.e., $|s| \leq \delta$) and at the same time $|s|$ is large enough so that the numerical processes we want to evoke are still well conditioned enough to yield the results we want to the accuracy we need (see Fig. 1). In a typical case of $z(t)$ converging to a singular solution as $t \to 0$, the condition number of the Jacobian matrix of $h(z(t), t)$ is blowing up and the number of digits of accuracy in the computed values for $z(t)$ is decreasing. Other numerical processes we want to evoke (e.g., curve fitting, prediction, correction) are being degraded concurrently as the round-off error increases. In practice, neither the upper bound defined by Theorem 1 nor the lower bound defined by round off is *a priori* available.

Finding the end game operating range has to do with discovering for what values of s the power series (1) is valid. Typically, this is done by advancing t from τ to 0 in steps and at each step evoking some test derived from (1), the test consisting of determining if some behavior predicted by (1) is occurring. (It is this feature of discovery that makes the end game *adaptive*.) Often, the pattern over steps of the test results will be used, and some sample of $z(t)$ is involved. One way to do this is to simply advance t to an arbitrary value (but perhaps based on previous numerical experience), conjecture that we are now in the range, and go on to parts B, C, and D. The success or failure of these later parts is then the "test." If we fail to obtain a satisfactory estimate of z^*, we then advance t further and try B, C, and D again. This is a valid and systematic scheme for determining the range (and carrying out the other Parts), although computationally wasteful. In 3.4 a more sophisticated approach is suggested based on the prediction error.

If z^* is nonsingular, then the operating range has a lower bound of zero, the nicest case. On the other hand, the operating range might be quite narrow, which will limit our options. For example, we might be forced to use the circular sample (Sect. 3.3). In fact, the operating range might be empty, in which case all our ways of implementing Theorem 1 will fail. This might be interpreted as implying that we need more digits of arithmetic to get the results we need.

For part B (estimating c), we might choose a sequence of test values and try each one by investigating if parts C and D can be completed successfully with that choice. We put forward several more efficient approaches, via the circular sample and the prediction error, in Sects. 3.3 and 3.4 below. In fact, each of these more effective ways of implementing Theorem 1 involves testing a limited range of c values one by one. However, this is not required for all approaches. For example, if we evoke Newton's method on a truncation of the power series (1) and solve for c along with the values for the coefficients, using some sample of $\mathcal{Z}(s)$, then we would not have to go through a range of values for c. In our tests of this approach, however, it worked well only when the alternatives worked better.

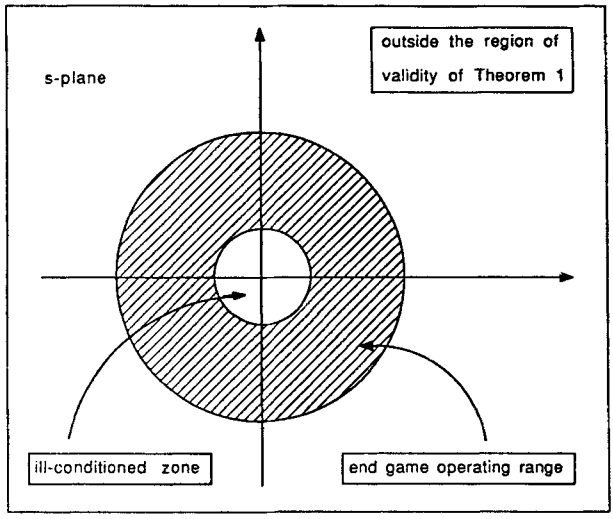

Fig. 1. Schematic illustrating the end game operating range

For part C we might sample $z(t)$ or $\mathcal{Z}(s)$ further before fitting the data to a truncation of (1). Sampling is discussed in Sect. 3.2.

Part D involves having a way of evaluating the quality of a numerical estimate of z^*. This evaluation is part of the adaptivity of the end game, in that certain parts are repeated (usually after t is advanced toward 0) until no further improvements in the evaluation of the estimate of z^* can be obtained. Part D is considered in Sect. 3.5.

3.2 Sampling the path

We have $z(t)$ defined as the unique path satisfying $h(z(t), t) = 0$, $z(1) = z^\sigma$, and $t \in [0, 1]$, while $\mathcal{Z}(s)$ is the unique power series defined by $\mathcal{Z}(s) = z(s^c)$ for some (smallest) positive c and $s \in [0, \delta)$ for some δ. Since we don't know c a priori, it is important to note that we can sample both with and without having an explicit value for c.

Consider the case that we do not have c. First, sample points, $z(t_j)$, can be taken with t_j positive real, and then later converted to $\mathcal{Z}(s_j)$ (after an estimate of c is obtained) via $\mathcal{Z}(s_j) = z(t_j)$ where $s_j = \sqrt[c]{t_j}$, and the root is the positive real root. Second, while $z(t)$ is strictly defined only for non-negative real t, we may continue $z(t)$ into the complex plane (that is, for non-real complex or negative t values) in a numerically straightforward way, obtaining new sample points $(\tilde{z}_j, \tilde{t}_j)$. (This is in the spirit of analytic continuation from complex variables.) Some care must be taken in converting $\tilde{z}_j$ to $\mathcal{Z}(s)$ to assign the correct c^{th} root of $\tilde{t}_j$ to s. The simplest scheme is to evoke a substitution $t \leftarrow r(\theta)e^{2\pi i\alpha(\theta)}$ where $r(\theta) > 0$ and $\alpha(\theta)$ are real-valued analytic functions, with $i \equiv \sqrt{-1}$. Then we can track the resulting homotopy continuation in θ. The resulting sample points $z(t_j)$ for $t_j = r(\theta_j)e^{2\pi i\alpha(\theta_j)}$ are then converted to $\mathcal{Z}(s_j)$ (after an estimate of c is obtained) via $s_j = \sqrt[c]{r(\theta_j)}e^{2\pi i\alpha(\theta_j)/c}$ where the root is the positive real root.

After we have a candidate value for c, we can use it to sample $\mathcal{Z}(s)$ directly. We can now predict and correct in s rather than t, and in fact we need not follow a path at

all to sample $\mathcal{Z}(s)$. Once we have established an approximation to (1) (via sampling and fitting with s in the end game operating range), we can predict by using this to obtain approximations to $\mathcal{Z}(s)$ for any s in the range and then correct in the usual way. To see the advantages of this, consider the following: Suppose we have some samples for $z(t)$ with t positive real and we wish to predict from these values across the origin to $z(t)$ for some negative real values of t. The singularity at $t = 0$ prevents us from doing this. However, in s we are not restricted. There is no singularity at $s = 0$ and we can (for example) predict to negative s points without difficulty.

3.3 The circular sample

The circular sample is an example of extending the continuation path via a substitution $t \leftarrow r(\theta)e^{2\pi i\alpha(\theta)}$, as discussed in the previous subsection. This sample is worthy of special consideration because it has some very powerful numerical properties. It is also closely related to the method of [16]. These facts are established in this subsection.

Suppose we choose $r(\theta) \equiv t_0$ for all θ, where t_0 is a constant positive real number in the end game operating range and $\alpha(\theta) \equiv \theta$. Then

$$z\left(t_0 e^{2\pi i\theta}\right) = \mathcal{Z}\left(\sqrt[c]{t_0}\, e^{2\pi i\theta/c}\right) \text{ for any } \theta$$

from which it follows that

$$z\left(t_0 e^{2\pi i\theta}\right) = z(t_0) \text{ when } \theta = c. \tag{4}$$

(Cf. Theorem 1 of [16].) Thus, we have a numerical way of discovering c, yielding at the same time a sample of $\mathcal{Z}(s)$. We call this *the circular sample*, and it generates high-order approximations to z^* as follows.

Take evenly spaced sample points of $z(t)$ as θ goes from 0 to c; more specifically, let n_0 be a positive integer greater than 1, $n_s \equiv n_0 c$, and $u \equiv e^{2\pi i/n_s}$. Take the samples

$$z\left(t_0 u^{j\cdot c}\right) = \mathcal{Z}\left(\sqrt[c]{t_0}\, u^j\right) \qquad \text{for } j = 1 \text{ to } n_s.$$

Note that u is a primitive n_s^{th} root of unity. It follows that $\sum_{j=1}^{n_s} u^j = 0$, and in fact $\sum_{j=1}^{n_s} u^{jk} = 0$, for any k that is not a multiple of n_s. Now from Theorem 1 we have

$$\begin{aligned} \frac{1}{n_s}\sum_{j=1}^{n_s} \mathcal{Z}\left(\sqrt[c]{t_0}\, u^j\right) &= \frac{1}{n_s}\left[\sum_{j=1}^{n_s} a_0 + \sum_{j=1}^{n_s} a_1\left(\sqrt[c]{t_0}\, u^j\right)\right. \\ &\qquad \left. + \sum_{j=1}^{n_s} a_2\left(\sqrt[c]{t_0}\, u^j\right)^2 + \quad \ldots \right] \\ &= a_0 + 0 + 0 + \quad \ldots \quad + a_{n_s}\left(\sqrt[c]{t_0}\right)^{n_s} + \quad \ldots \end{aligned} \tag{5}$$

Thus, the average of the sampled points, $\frac{1}{n_s}\sum_{j=1}^{n_s} z(t_0 u^{j\cdot c})$, approximates $z^* = a_0$ up to the n_s^{th} term of the power series defined in Theorem 1. This is the integral formula

$$z^* = \frac{1}{2\pi c}\int_0^{2\pi c} z\left(t_0 e^{2\pi i\theta}\right) d\theta$$

from Theorem 1 of [16] when the integral is approximated by the rectangle rule. From (5) we can see how this sample (and the related integral formula) can easily yield high-accuracy solution estimates. Note also that it will work even if the end game operating range is very narrow. However, it is (relatively) expensive to compute, since continuing the path around the origin might take as much (or more) computational work as tracking the main part of the path. Thus we view the circular sample as a last resort when cheaper samples fail to give us the desired accuracy.

Note that to use the circular sample in a computer code we must enter a maximum c test bound, say $c_{\max}$. Each time θ is advanced to an integer value, $\hat{c}$, we test $\hat{c}$ to see if it is c (via Eq. (4) above). We test from $\hat{c} = 1$ to $\hat{c} = c_{\max}$. If $c_{\max}$ is set too small, then (4) will not be satisfied.

If t_0 is not in the end game operating range, then (4) might be satisfied for some $\hat{c}$, but the resulting solution estimate will probably not be a good approximation to z^* and will fail the evaluation part of the end game (Part D).

For large c values, we have found nothing better than the circular sample[2]. For small c values, the method described in the next subsection is generally just as accurate and cheaper to compute.

3.4 The prediction error

We can use the prediction error to find the end game operating range adaptively and to estimate c. The following discussion clarifies what is involved.

When we enter the end game, we predict by sampling $z(t)$ and fitting these points to (1). There will be errors in this process that we summarize as ϵ_P, the *prediction error*. We recognize two components of ϵ_P. First, there is *round-off error*, ϵ_R. Second, to fit (1) we truncate to an order v polynomial, introducing *truncation error*, ϵ_T.

After we enter the end game operating range, t will be small enough for Theorem 1 to hold, and, as t gets smaller, e_T is eventually dominated by the first omitted term of (1)[3]. Thus

$$\epsilon_T \approx a_{v+1} t^{(v+1)/c}$$

and taking the logarithm we have

$$\text{Log}(\epsilon_T) \approx \text{Log}(a_{v+1}) + (v+1)/c \ \text{Log}(t), \tag{6}$$

giving the slope on a log-log plot of e_T versus t as $(v+1)/c$. If e_R is still dominated by e_T, the slope of e_P becomes $(v+1)/c$ as predicted by (6) for e_T. Now, ϵ_P can be estimated numerically, as described below, so one test for the end game operating range is that the computed slope look like $(v+1)/c$ for some c.

Now suppose we try to fit $z(t)$ to a series of the form of (1) but with an incorrect choice of c, say $\hat{c}$. Then the slope of the prediction error will be different, because the number of terms in the series with $\hat{c}$ that match terms in (1) will be less than v. If $\hat{c}$ is not a multiple of c, then the slope will be $1/c$. If $\hat{c} = uc$, where $u > 1$ is an integer, then the slope will be $(\text{Int}(v/u)+1)/c$ where Int denotes the greatest integer. In either

[2] If the system we are solving is *polynomial* in addition to being analytic, then the method of [17] can work well for large c.

[3] This discussion assumes that all coefficients in (1) are non-zero. For generically constructed homotopies, this is a reasonable hypothesis. For non-generically constructed homotopies, it may not hold and a more detailed analysis of cases may be required.

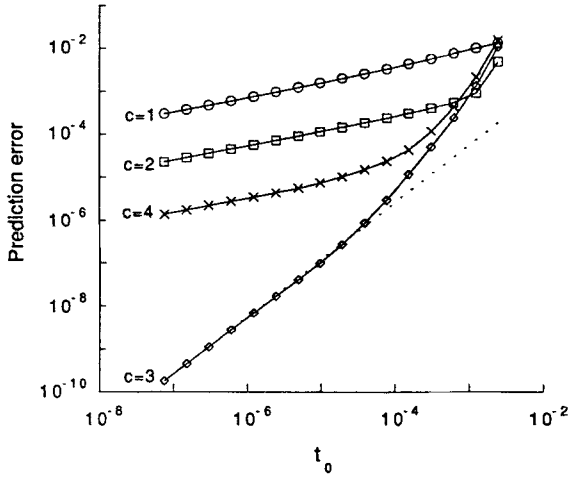

Fig. 2. Determination of cycle number c for Problem 1

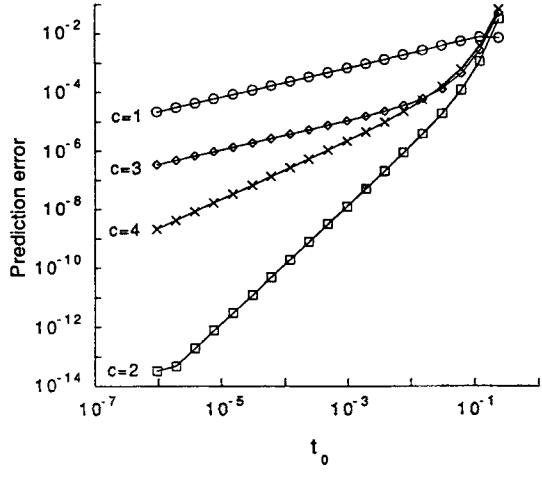

Fig. 3. Determination of cycle number c for Problem 2

case, the correct c will tend to have a markedly smaller associated prediction error. Therefore, the *magnitude* of the prediction error (rather than the *slope*) can also be a test for the correct c. Figures 2 and 3 show the dramatic difference for the correct c and several incorrect candidate values.

Now, in the typical singular case, as $t \rightarrow 0$, eventually e_P will be dominated by e_R. What we see in numerical tests is that e_P levels off and, if the path is not terminated, it may even increase. This point of leveling off marks the end of the end game operating range. (This is suggested in Fig. 3 where the c=2 curve is just beginning to flatten out.)

One source of error we have not discussed is *fitting error*, the error introduced in the estimated coefficients by the numerical method used to fit the sample. With the

simple (linear) fitting methods we use, the fitting error is essentially accounted for via e_R.

Now, e_P can be computed numerically by a scheme such as the following. Sample $v+2$ points, fit using $v+1$ of the points and predict the $(v+2)^{th}$ point. Let the difference between the predicted and actual be the numerical e_P. It makes the most sense as $t \to 0$ to let the latest sample point be the $(v+2)^{th}$. Figures 2 and 3 are plots of such numerical e_P values. See the next Section for the exact details.

3.5 Evaluating solution estimates

Let z^+ be an estimate of the singular solution z^*. Simply looking at $|f(z^+)|$ to evaluate the quality of the estimate can be misleading, because $|f(z^+)|$ is not scale independent. If z^* were nonsingular (and well conditioned), we could evaluate the quality of the approximation via the Newton's method residual. However, this residual is not so accurate for approximations of singular z^* and tends to blow up as $t \to 0$. Using the power series (1) we can develop a much better-behaved error formula which has some of the desirable qualities of the Newton's method residual (e.g., it is scale independent). However, if we are not in the end game operating range, and, in particular, if we have already entered the zone of significant round-off, then this formula will be degraded. We usually track the error as $s \to 0$, noting it (typically) decreasing to a minimum, then leveling off or increasing. The s value associated with the minimum may be taken as indicating the inner border of the end game operating range and the associated solution estimate will be (essentially) the best estimate we can obtain.

We develop this error formula in the context of tracking the path converging to z^*, as follows: For a fixed order polynomial, we fit sample data points from the continuation path for smaller and smaller s values. Each fit of the data to the polynomial provides an estimate of z^* (namely, the constant term). We get a good approximation to the error in the estimates by considering differences of successive estimates divided by a factor generated from Richardson extrapolation. We illustrate by developing the following explicit case, which we use in Sect. 4.

Let p be a positive number greater than 1. Consider a sample taken at s_2 and $s_1 = s_2/p$ and a succeeding sample at s_1 and $s_0 = s_1/p$. We sample both $\mathcal{Z}(s)$ and $\mathcal{Z}'(s)$, where $\mathcal{Z}'(s)$ represents the derivative of $\mathcal{Z}(s)$ with respect to s. (It is easy to get first derivatives from the continuation process.) Each sample determines a cubic:

$$\mathcal{Z}(s) = b_0 + b_1 s + b_2 s^2 + b_3 s^3.$$

Let $b_{0,2}$ denote the constant coefficient developed from the $\{s_2, s_1\}$ sample, and $b_{0,1}$ the constant coefficient developed from the $\{s_1, s_0\}$ sample. By examining the fitting process, we derive a formula

$$a_0 - b_{0,j} = p^2 a_4 s_j^4 + O\left(s_j^5\right)$$

for $j = 1, 2$, where a_0 and a_4 are terms of (1). Then by Richardson extrapolation we get

$$\frac{|b_{0,1} - b_{0,2}|}{p^4 - 1} = |b_{0,1} - z^*| + O\left(s_1^5\right), \tag{7}$$

using $z^* = a_0$. The left hand side of (7) is the (computable) error formula that we track as $s \to 0$.

4. An implementation and numerical experiments

Our main path tracker is from the CONSOL8 code of [10]. Here we give a description of the adaptive end game, which is based on the prediction error approach of Sect. 3.4. After that, the two numerical experiments are presented. The homotopies are described with the experiments.

The end game proceeds in steps, with t decreasing at each step. For each step, we have sample points $z(t)$ for several different values of t. We test values $\hat{c}$ from $1, \ldots, c_{\max}$. For each $\hat{c}$, we convert to sample values $(\mathcal{Z}(s), s) = (z(t), t^{1/\hat{c}})$, fit a power series to these values, and test the prediction error. The value of $\hat{c}$ that gives the smallest error is taken as our guess of c.

While we could sample at both real and complex values of t, for simplicity, we have chosen real values only. This is particularly convenient, of course, when the homotopy has been programmed in real arithmetic. In the first step, we take samples at $t_2 = \tau$, $t_1 = \tau/r$, $t_0 = \tau/r^2$, where we take $r = 2$, $c_{\max} = 4$, and $v = 3$, where v is the order of truncation that defines the prediction error. The points $z(t_i)$ are found using the main path tracker, and the derivatives dz/dt are also saved for each sample point. (On subsequent steps, we use the best power series fit so far to predict the next point. This is more efficient than the standard predictor of the path tracker, which assumes $c = 1$.) For each trial value $\hat{c}$, we convert the sample points to s-space as $(\mathcal{Z}(s_i), s_i) = (z(t_i), t_i^{1/\hat{c}})$, $d\mathcal{Z}/ds = \hat{c} s_i^{\hat{c}-1} dz/dt$, fit a cubic in s to the values and derivatives at t_2, t_1, evaluate the cubic at s_0 to predict $\mathcal{Z}(s_0)$, and compute the difference $\mathcal{Z}(s_0) - z(t_0)$. The norm of this difference is the prediction error. If none of the fits are accurate, skip to the next step by advancing t_0 to t_0/r and updating (t_2, t_1, t_0) to $(t_1, t_0, t_0/r)$.

If we have not skipped to the next step, then we have determined that we are in the end game operating range with an acceptable estimate of c. Now we must obtain a sufficiently accurate estimate of z^*.

The sample values $\mathcal{Z}(s_i)$, $i = 0, 1, 2$ and their derivatives are already available. We could choose to use only these values or to collect a larger sample. The larger sample can be taken at real or complex values of s. In the numerical examples below, we show the results of two strategies: using the original samples alone and doubling the number of sample points to include $\mathcal{Z}(s)$ at $s = -s_i$, $i = 0, 1, 2$. Note that even though we cannot sample $\mathcal{Z}(0)$ directly, because the Newton corrector is unreliable at the singularity, we can jump across the origin to negative values of s. We use a cubic prediction from s_1, s_0 to $s = -s_0$, and correct with Newton's method, and proceed similarly out to $-s_1, -s_2$. Note that a real path for $s > 0$ always continues to a real path for $s < 0$, which is convenient when working only in real arithmetic. For the prediction across the origin to be successful, it is critical to have the correct value of c.

To estimate $z^* = \mathcal{Z}(0)$ and its error, we compute an estimate z_2^+ from the values at $\pm s_1, \pm s_2$ and an estimate z_1^+ from $\pm s_0, \pm s_1$. In the case where we use only positive s values, $\mathcal{Z}(0)$ is obtained from a cubic fit to the points and their derivatives. Since $s_1 = r^{1/c} s_0$, the error in z_1^+ is approximately $|z_1^+ - z_2^+|/(r^{4/c} - 1)$ from (7). In the case where we include the negative s values, we can increase the order of the fit to 7. Then the error in z_1^+ is $|z_1^+ - z_2^+|/(r^{8/c} - 1)$ by reasoning similar to that used to develop (7).

Two details of the numerical fit are worth mentioning. First, to maintain stability of the fit as $s_0 \to 0$, it is helpful to rescale s. We rescale by the larger s_i value, so

that the largest scaled coordinate is 1. Second, in the higher order fit, we use the fact that we have sampled symmetrically about $s = 0$. Considering the power series (1), we find that

$$(\mathcal{Z}(s) + \mathcal{Z}(-s))/2 = \sum_{k=0}^{\infty} a_{2k} s^{2k},$$

that is, the odd terms drop out. Therefore, since we only want to compute $\mathcal{Z}(0) = a_0$, we may compute the order 7 result by averaging the symmetrical data points, converting the abscissa to $w = s^2$, and fitting a cubic in w.

If the error estimate for z^* is larger than the desired accuracy, we take another step by advancing t_0 to t_0/r and proceed as above to estimate z^* with the new sample. We continue until the error estimate meets the specified accuracy or the prediction error test indicates that we have exited the end game operating range.

If the specified accuracy is never met, we then report the estimated end point with the smallest error, and flag the lack of full convergence. It may be that either the path became ill-conditioned before an accurate estimate could be computed or simply that the correct cycle number is larger than $c_{\max}$.

We report now on the results of solving two test problems: one polynomial and one non-polynomial.

Problem 1. Griewank and Osborne's problem

The system

$$f_1 = (29/16)z_1^3 - 2z_1 z_2$$
$$f_2 = z_2 - z_1^2$$

is taken from [8], p. 749. It is notable in that Newton's method diverges for start points near the solution $(x, y) = (0, 0)$. We used the homotopy

$$h(z, t) = (1 - t)f(z) + t\gamma g(z),$$

where the start system $g(z)$ is

$$g_1 = z_1^3 - 2^3$$
$$g_2 = z_2^2 - 2^2$$

and $\gamma = 0.123247542 + \mathrm{i}\,0.76253746298$. By matching the degrees of f and g and picking γ as a "random" complex number, we are assured that the homotopy obeys the required conditions [12]. By construction, the start point $z = (2, 2)$ satisfies $h(z, 1) = 0$. As $t \to 0$, the path $z(t)$ approaches the singular solution at $z = (0, 0)$.

Figure 2 shows the log-log plots of prediction error versus t_0 for different candidate cycle numbers. The correct cycle number $c = 3$ quickly emerges, decreasing at a much faster rate than the other trial values. In fact, the slope is 4/3 for $\hat{c} = 3$ and 1/3 for $\hat{c} = 1, 2, 4$. (A slope of exactly 4/3 is indicated by the dotted line.) This is the expected result for a cubic predictor and $c = 3$ (see Sect. 3.4 above).

Table 1 shows a corresponding table of end point error versus t_0. The first column of error data shows the result of a simple cubic extrapolation in t, which is equivalent to fixing $\hat{c} = 1$. The last two columns use the (correct) cycle number $c = 3$, with

Table 1. Error in endpoint estimates for Problem 1

	$\hat{c} = 1$	$\hat{c} = 3$	
t_0	Order = 3	Order = 3	Order = 7
0.7813e − 04	0.2239e + 00	0.1283e − 03	0.8622e − 07
0.3906e − 04	0.1719e + 00	0.4318e − 04	0.1346e − 07
0.1953e − 04	0.1329e + 00	0.1778e − 04	0.2109e − 08
0.9766e − 05	0.1032e + 00	0.7284e − 05	0.3312e − 09
0.4883e − 05	0.8054e − 01	0.2967e − 05	0.5207e − 10
0.2441e − 05	0.6305e − 01	0.1202e − 05	0.8192e − 11
0.1221e − 05	0.4949e − 01	0.4850e − 06	0.1289e − 11
0.6104e − 06	0.3894e − 01	0.1950e − 06	0.2031e − 12
0.3052e − 06	0.3069e − 01	0.7822e − 07	0.3199e − 13
0.1526e − 06	0.2422e − 01	0.3130e − 07	0.5113e − 14
0.7629e − 07	0.1914e − 01	0.1250e − 07	0.8316e − 15

cubic and seventh-order fits, respectively. The errors decrease as $t_0^{1/3}$, $t_0^{4/3}$, and $t_0^{8/3}$, respectively, in accordance with the expected result.

Table 1 lists the actual error, since in this test example we know the solution. In the general case, we would not know the actual error and would have to depend upon the error estimate to judge when to stop the iterations. In light of this, it is useful to note that the error estimate based on Richardson extrapolation is accurate to within 10% until round-off error begins to dominate near $t = 0$ (i.e., when the error approaches the precision limit, about 10^{-13} in double precision). At that point, the raw difference $|z_1^+ - z_2^+|$ is a better indicator.

Problem 2. A transcendental problem

The system

$$f_1 = e^{-z_1^2} + e^{-z_2^2} - 2$$
$$f_2 = \sin(z_1) + \sin(z_2)$$

has a singular solution at $z = (0, 0)$. We chose the homotopy

$$h(z, t) = f(z) - tf(z^\sigma),$$

where z^σ is the start point for $t = 1$. We used $z^\sigma = (1, 0.5)$. Note that this problem is *not* polynomial. Moreover, the homotopy path is *real*. We did not know a priori that the homotopy would satisfy condition 3 of Sect. 2; that is, there is nothing inherent in the construction to guarantee that the homotopy path would not be singular somewhere for $t \in (0, 1]$. However, the success of the method shows that this did not happen.

Figure 3, similar to Fig. 2, shows the prediction error versus t_0. Again, the correct cycle number $c = 2$ is clear. The slopes on the log-log plot are 1/2 for $\hat{c} = 1, 3$, whereas for $\hat{c} = 2$ it was $4/2 = 2$ and for $\hat{c} = 4$ it was $(\text{Int}(3/2) + 1)/2 = 1$. Of course, we prefer to use the $\hat{c} = 2$ prediction, because it converges much more rapidly. The leveling off of the $\hat{c} = 2$ line around $t_0 = 10^{-6}$ is due to the onset of round-off error as the prediction approaches full double-precision accuracy.

Table 2 reports the error in the end point estimates. The rates of decrease are in close agreement with the expected results of $O(t_0^{1/2})$, $O(t_0^{4/2})$, and $O(t_0^{8/2})$, until

Table 2. Error in endpoint estimates for Problem 2

	$\hat{c} = 1$	$\hat{c} = 2$	
t_0	Order = 3	Order = 3	Order = 7
0.1563e − 01	0.1635e + 00	0.4079e − 04	0.3672e − 08
0.7813e − 02	0.1155e + 00	0.9804e − 05	0.2235e − 09
0.3906e − 02	0.8167e − 01	0.2403e − 05	0.1379e − 10
0.1953e − 02	0.5774e − 01	0.5950e − 06	0.8562e − 12
0.9766e − 03	0.4083e − 01	0.1480e − 06	0.5334e − 13
0.4883e − 03	0.2887e − 01	0.3692e − 07	0.3323e − 14
0.2441e − 03	0.2041e − 01	0.9218e − 08	0.1167e − 13
0.1221e − 03	0.1443e − 01	0.2303e − 08	0.2831e − 13
0.6104e − 04	0.1021e − 01	0.5756e − 09	0.2340e − 14
0.3052e − 04	0.7217e − 02	0.1439e − 09	0.4643e − 13
0.1526e − 04	0.5103e − 02	0.3597e − 10	0.1410e − 13
0.7629e − 05	0.3608e − 02	0.8992e − 11	0.7818e − 13
0.3815e − 05	0.2552e − 02	0.2248e − 11	0.8411e − 13
0.1907e − 05	0.1804e − 02	0.1234e − 11	0.1640e − 13

round-off error stalls the cubic at error $\approx 0.2 \times 10^{-11}$ and stops the seventh order fit at error $\approx 0.3 \times 10^{-14}$. Doggedly pushing t_0 beyond such a point is not only useless, but may actually increase the error in the end point. Again, the table lists the actual error and the estimated error was very close. When the estimated error levels off, the iterative process should be stopped. For comparison, we note that Newton's method initiated from the same start point $(1, 0.5)$ converges to accuracy $\approx 0.4 \times 10^{-8}$. Such degraded accuracy is common for Newton's method near a singularity.

5. Application to engineering problems via polynomial continuation

The new ideas for computing singular solutions to analytic systems presented in this paper and in [16, 17] were originally developed to deal with the large number of singular endpoints encountered when polynomial continuation [10, 12–15] is used to solve certain kinds of polynomial systems that arise in several engineering areas [9,11,21, 23, 24]. In this section, we describe briefly the context of polynomial continuation and then report on the solution of two problems of significant difficulty from the area of the kinematics of mechanisms.

Sometimes in applications one wishes to compute the full set of solutions to a polynomial system. Usually, there will be a subset of physical significance which can be gleaned from this full solution set (the real solutions, the real solutions with non-negative components, the finite solutions, etc.). The method of *polynomial continuation* can be used to compute the geometrically isolated solutions in the full solution set. To obtain a start system one constructs a system which is easily solvable and which captures the generic structure of the target system. Then, each solution of the start system is taken to be the start point for a continuation path whose end point will be a solution to the target system. This set of endpoints contains the set of geometrically isolated solutions to the target system. Even though one would often be content to have only the nonsingular solutions, when using polynomial continuation there is

no way to track just the paths with nonsingular endpoints. Dispensing with the singular endpoints quickly and efficiently becomes the major expense of the computation. One must compute the singular endpoints accurately enough to know that it is safe to discard them. For a general description of polynomial continuation, see the above cited references.

In evoking polynomial continuation for the following problems, we used the same path tracker and adaptive end game described in the previous section, with the multi-homogeneous start system from [22]. In polynomial continuation, the number of paths that is tracked is equal to the generic number of solutions associated with the structure of the target system. However, this generic number is different depending on the space in which the solutions are envisioned as belonging to. In complex projective space, the number of paths is simply the total degree of the system (the product of the degrees of the equations). However, it can be advantageous to view the solutions as belonging to other spaces. If the system is naturally *multi-homogeneous*, then the appropriate space is a product of projective spaces, and the number of paths will be the *Bezout number*. These ideas were first used in the context of polynomial continuation in [12], and further developed in [13–15]. The tutorial [24] gives an overview of the multi-homogeneous approach to polynomial continuation. For the following two problems, a 2-homogeneous structure is used.

Problem 3. Inverse kinematics, model 1

We choose as our first application the second test problem from [21]. It's significance to mechanical engineering is described there, and the physical solutions are given. This is a system of eight second-degree equations in eight unknowns. The problem was analyzed as a 2-homogeneous system in [12] and [13], where it was shown that the Bezout number is 96. Further, the 2-homogeneous generic solution structure of this system was proven (in Theorem 4 of [13]) to consist of 64 nonsingular solutions and 8 multiplicity-four (geometrically isolated) singular solutions. See [13] for a complete mathematical description of the system.

Interestingly, even though the singular solutions have multiplicity 4, the associated path end points all have cycle numbers of 2.

We ran three parallel runs, to get some comparisons for the *success* and the *cost* of the power series end game. Each path was tracked from $t = 1$ to $t = 10^{-3}$. This was designated the "main part of the path." Then each of three end games was evoked. First, we ran the power series end game. Second, we substituted a Newton's method end game, whereby at the designated point where the end game was begun, we evoked a Newton's method. Third, we ran the code "without end game," that is, each path was tracked until it converged with $t = 0$ or ill conditioning caused the path tracking to be terminated before $t = 0$. (For the nonsingular end points, convergence to $t = 0$ was always achieved.)

The Newton's method end game is not too unreasonable for paths converging to geometrically isolated solutions, but it is risky, since the Newton's method (essentially) abandons the path and need not converge to the path end point or converge at all. Naturally, many methods for computing singular solutions to polynomial systems could be substituted for the Newton's method, to produce new end games.

The "end game" consisting of tracking the path until it either converges or fails due to ill conditioning is more conservative than the Newton's method end game, particularly with the path tracker we are using (from CONSOL8 [10]). It respects the basic geometry of the path and often computes the end point to several digits

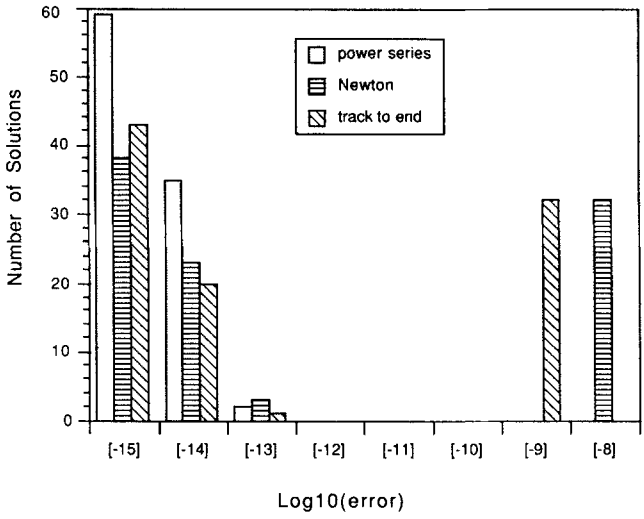

Fig. 4. Comparing the success of three end games on Problem 3. The abscissas give the exponents of the errors of path end points. The ordinates count the number of end points with the designated errors

of accuracy even when the path is converging to a positive-dimensional solution set. However, it tends to use a large number of steps when the end point is singular.

The *success* of the various end games is summarized in Fig. 4, which gives the errors of the 96 computed endpoints. (These are actual relative errors, generated with reference to a solution set computed in extended precision. The abscissa values are the exponents of the error; thus, -14 denotes an error of 10^{-14}. The ordinates gives the number of endpoints with the designated error.) Observe that the power series end game was very successful on this problem, in that all the endpoints were computed to good double precision accuracy, while the other two methods produced single precision accuracy on the singular solutions.

The *costs* of the end games are given in Table 3 in terms of NFE and CPU. (NFE stands for "number of function evaluations." It is approximately equal to the number of 8×8 linear systems solved and is a good relative measure of computational work. CPU stands for the cpu time in minutes on an IBM 360-3090.) The cost of the main part of the path tracking is included for comparison. The power series method was the most expensive, but it was the only one to work really well. It cost about a third of the cost of the main part of the path tracking.

Table 3. Work Statistics for Problem 3. NFE denotes the number of function evaluations. CPU denotes the cpu time in minutes on an IBM 360-3090

	NFE	CPU
Main Path Tracking	32621	2.6
Power End Game	9735	0.8
Newton End Game	1214	0.1
Tracked to End	5051	0.4

Problem 4. Inverse kinematics, model 2

Our second application is the system defined in [23] by Eqs. (14)–(16) with coefficients given in Table 1 of that reference. It's significance to mechanical engineering is described there. This is a system of ten second-degree equations and two linear equations in twelve unknowns. In [23] the problem is analyzed as a 2-homogeneous system, and it was shown to have a Bezout number of 320. It has 16 nonsingular solutions. By computation, it was determined that the rest of the solutions are singular (not necessarily geometrically isolated). In solving this system using a 2-homogeneous polynomial continuation, 16 of the paths will converge to the sixteen nonsingular solutions, and the rest of the paths will converge to nonsingular endpoints.

The same test arrangement with three end games was used on this problem as for Problem 3. The results are summarized in Table 4 and Fig. 5. Each solution has either cycle number 1 or 2. (There are exactly 16 nonsingular solutions and 16 cycle-number-one singular solutions.) Again, the power series method worked by far better than the others, for a cost of about a third of the cost of the main path tracking. Most endpoints were obtained to at least 10 digits of accuracy, although 27 out of 320 were not.

Table 4. Work Statistics for Problem 4. NFE denotes the number of function evaluations. CPU denotes the cpu time in minutes on an IBM 360-3090

	NFE	CPU
Main Path Tracking	145 884	16.0
Power End Game	42 061	4.9
Newton End Game	5 206	0.3
Tracked to End	61 670	7.0

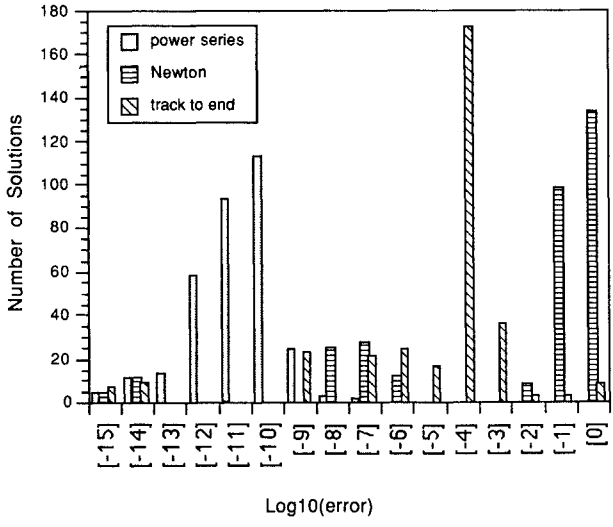

Fig. 5. Comparing the success of three end games on Problem 4. The abscissas give the exponents of the errors of path end points. The ordinates count the number of end points with the designated errors

While we considered this a very satisfactory result for such a challenging problem, we did investigate computing the 27 "bad" paths to better accuracy, via the integral method of [16]. In our implementation, this is exactly the same as using the circular sample and averaging, as described in Sect. 3.3. This improved the accuracy of the "bad" solution set considerably: 17 had an error of -12, 6 of -11, 3 of -10, and 1 of -9. This required about 38,000 function evaluations. Thus, the work required to improve these 27 end points using the circular sample was almost equal to the work required by the power series method to generate estimates for all 320 end points. Part of the reason for this is that our implementation of the circular sample adaptively seeks to find a t_0 value where the best residual is produced. In some cases, this requires that a number of circular samples be taken for a variety of t_0 values. We conclude there is a very narrow end game operating range for these paths, but wide enough for the circular sample, if a suitable t_0 value is discovered.

6. Summary and conclusions

In this paper we have presented a power series method for computing accurate approximations to singular solutions of analytic systems. The method works in either real or complex arithmetic. A homotopy must be constructed with a path converging to the solution, and a numerical method for tracking the path must be available. A theorem is proven which shows that a power series exists for the path in a neighborhood of the solution. (The power series is expressed in terms of the cycle-number root of the given path parameter.) The numerical method approximates this power series and then takes the constant term as an estimate of the solution. The performance of the method on two simple test examples is discussed in detail. The results of solving two challenging polynomial problems from the kinematics of mechanisms are given.

Disadvantages of the method include the need to set up the homotopy and path tracking machinery. Although this machinery is well developed, it may seem awkward to those who are new to it. The method works best when c is small. For larger c, implementations relying on polynomial fitting tend to need extended precision. While the circular sample approach doesn't require fitting and works well for larger c, it is relatively expensive.

Advantages of the method are that it is easy to program in a path tracking context, and if that context is *real* rather than complex, no conversion to complex arithmetic is needed. Also, the method requires no conditions on the system, save that it be analytic, nor conditions on the rank of the Jacobian matrix at the solution. The numerical experiments show that the estimates of the solution can be dramatically better than that obtained by standard path tracking or Newton's method.

Although the theoretical basis of the method is well established by Theorem 1, further numerical studies, especially on problems generating larger c values, would be helpful to clarify the practical limits of the method and to suggest improvements in implementation.

Acknowledgement. This work represents a cooperative interdisciplinary effort. The order of the names given on this paper has no significance. There is no primary author.

We would like to thank Dr. Daniel Baker of the General Motors Research Laboratories Mathematics Department for several helpful discussions.

Appendix

Proof of Theorem 1. We recall some results from section A-2 of [16]. (The variable "t" here is equal to "$1 - t$" there.) There is a neighborhood D' of z^* and an $\epsilon > 0$ such that

a) the restriction, $p_{K'}$, of the projection $D' \times \Delta_\epsilon \to \Delta_\epsilon$ to $K' = K \cap (D' \times \Delta_\epsilon)$ is proper and c is the local degree of the proper map, $p_{K'}$, at z^*;
b) letting $\delta = e^{1/c}$ and $D_\delta \equiv \{s \in C^1 \mid |s| < \delta\}$, there is a one to one holomorphic map $\nu : D_\delta \to K'$ with $\nu(0) = z^*$, such that $\alpha \equiv p \circ \nu$ is given by $t = s^c$;
c) for $t \in [0, \epsilon]$, $z(t) \in \nu(D_\delta)$.

Define the vector function $\mathcal{Z}(s) \equiv z \circ \nu$. It follows that that

$$z(s^c) = \mathcal{Z}(s)$$

for $s \in [0, \delta]$. Since $\mathcal{Z}(s)$ is holomorphic on D_δ, there is a convergent power series expansion,

$$\mathcal{Z}(s) = \sum_{k=0}^{\infty} a_k s^k \text{ on } D_\delta.$$

It remains to prove the minimality property of c. Assume that there exists a positive integer c', a $\delta' > 0$, and a holomorphic vector function $\mathcal{Z}'(s')$ defined on $D_{\delta'}$, and with

$$z(s'^{c'}) = \mathcal{Z}'(s') \text{ for } s' \in [0, \delta'].$$

We must show that $c' \geq c$.

To see this note that

$$h\left(\mathcal{Z}'(s'), s'^{c'}\right) = h\left(z(s'^{c'}), s'^{c'}\right) = 0 \text{ for } s' \in [0, \delta'].$$

Thus $h(\mathcal{Z}'(s'), s'^{c'})$ is identically 0 since it is holomorphic. This shows that by making δ' smaller we have that the image of $D_{\delta'}$ under $\nu' \equiv (\mathcal{Z}'(s'), s'^{c'})$ lies on K' with $\nu'(0) = z^*$. The degree of the composition $p_{K'} \circ \nu'$ at 0 is at least the degree of the map $p_{K'}$ at $z^* = \nu'(0)$. Since the degree of $p_{K'} \circ \nu' = s'^{c'}$ is c', and the degree of $p_{K'}$ at z^* is c, we have that $c' \geq c$.

This completes the proof of Theorem 1.

References

1. Allgower, E.G., Georg, K. (1980): Simplicial and continuation methods for approximating fixed points and solutions to systems of equations. SIAM Rev. **22**, 28–85
2. Decker, D.W., Keller, H.B., Kelley, C.T. (1983): Convergence rates for Newton's method at singular points. SIAM J. Numer. Anal. **20**, 296–314
3. Decker, D.W., Kelley, C.T. (1980): Newton's method at singular points, I. SIAM J. Numer. Anal. **17**, 66–70
4. Decker, D.W., Kelley, C.T. (1980): Newton's method at singular points, II. SIAM J. Numer. Anal. **17**, 465–471
5. Decker, D.W., Kelley, C.T. (1985): Broyden's method for a class of problems having singular Jacobian at the root. SIAM J. Numer. Anal. **22**, 566–574
6. Garcia, C.B., Zangwill, W.I. (1981): Pathways to Solutions, Fixed Points, and Equilibria. Prentice-Hall, Englewood Cliffs, N.J.

7. Griewank, A. (1985): On solving nonlinear equations with simple singularities or nearly singular solutions. SIAM Rev. **27**, 537–563
8. Griewank, A., Osborne, M.R. (1983): Analysis of Newton's method at irregular singularities. SIAM J. Numer. Anal. **20**, 747–773
9. Meintjes, K., Morgan, A.P. (1987): A methodology for solving chemical equilibrium systems. Appl. Math. Comput. **22**, 333–361
10. Morgan, A.P. (1987): Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems. Prentice-Hall, Englewood Cliffs, N.J.
11. Morgan, A.P., Sarraga, R.F. (1981): A method for computing three surface intersection points in GMSOLID. ASME paper 82-DET-41 (1982). Also available as GMR-3964, G.M. Research Laboratories, Warren, MI
12. Morgan, A.P., Sommese, A.J. (1987): A homotopy for solving general polynomial systems that respects m-homogeneous structures. Appl. Math. Comput. **24**, 101–113
13. Morgan, A.P., Sommese, A.J. (1987): Computing all solutions to polynomial systems using homotopy continuation. Appl. Math. Comput. **24**, 115–138
14. Morgan, A.P., Sommese, A.J. (1989): Coefficient-parameter polynomial continuation. Appl. Math. Comput. **29**, 123–160
15. Morgan, A.P., Sommese, A.J. (1990): Generically nonsingular polynomial continuation. In: E. Allgower, K. Georg, eds., Lectures in Applied Mathematics, Vol. 26. Computational Solution of Nonlinear Systems of Equations. American Mathematical Society
16. Morgan, A.P., Sommese, A.J., Wampler, C.W. (1991): Computing singular solutions to nonlinear analytic systems. Numer. Math. **58**, 669–684
17. Morgan, A.P., Sommese, A.J., Wampler, C.W. (1992): Computing singular solutions to polynomial systems. Advances Appl. Math. (to appear)
18. Rall, L.B. (1966): Convergence of the Newton process to multiple solutions. Numer. Math. **9**, 23–37
19. Reddien, G.W. (1978): On Newton's method for singular problems. SIAM J. Numer. Anal. **15**, 993–996
20. Rheinboldt W.C., Burkardt, J.V. (1983): Algorithm 596: A program for a locally parameterized continuation process. ACM Trans. Math. Software **9**, 236–241
21. Tsai, L.-W., Morgan, A.P. (1985): Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods. ASME J. Mech., Transm., Autom. Design **107**, 189–200
22. Wampler, C.W. (1992): An efficient start system for multi-homogeneous polynomial continuation. Submitted for publication
23. Wampler, C.W., Morgan, A.P. (1991): Solving the 6R inverse position problem using a generic-case solution methodology. Mech. Machine Theory **26**, 91–106
24. Wampler, C.W., Morgan, A.P., Sommese, A.J. (1990): Numerical continuation methods for solving polynomial systems arising in kinematics. ASME J. Design **112**, 59–68
25. Watson, L.T. (1979): A globally convergent algorithm for computing fixed points of C^2 maps. Appl. Math. Comput. **5**, 297–311
26. Watson, L.T., Billups, S.C., Morgan, A.P. (1987): HOMPACK: A suite of codes for globally convergent homotopy algorithms. ACM Trans. Math. Software **13**, 281–310