Contemporary Mathematics
Volume **556**, 2011

A Parallel Endgame

Daniel J. Bates, Jonathan D. Hauenstein, and Andrew J. Sommese

Abstract. Numerical algebraic geometry is the area devoted to the solution and manipulation of polynomial systems by numerical methods, which are mainly based on continuation. Due to the extreme intrinsic parallelism of continuation, polynomial systems may be successfully dealt with that are much larger than is possible with other methods. Singular solutions require special numerical methods called endgames, and the endgames currently used do not take advantage of parallelism. This article gives an overview of continuation and endgames in the context of parallel computation. We also introduce a novel parallel algorithm for performing endgames at the end of homotopy paths, based on the Cauchy endgame, along with some heuristics useful in its implementation. This method, which has been implemented in the Bertini software package, leads to a significant gain in efficiency.

1. Introduction

Numerical algebraic geometry is the area devoted to the solution and manipulation of polynomial systems by numerical methods, which are mainly based on continuation. The methods of this field have been used to address numerous problems in many areas of science and engineering, as well as algebraic geometry.

Almost all of the computational effort of the algorithms of numerical algebraic geometry [**17**] reduce to the following core computation:

> Given a system $f(x) = 0$ of N polynomials in N unknowns, continuation computes a finite (multi)set S of solutions such that:
> - any isolated root of $f(x) = 0$ is contained in S; and
> - any isolated root appears a number of times equal to its multiplicity as a solution of $f(x) = 0$.
>
> The set S is called a superset of the isolated solutions as it may contain solutions which are not isolated.

2010 *Mathematics Subject Classification.* 14D05, 14Q99, 65H10, 65H20, 65E05, 65Y05.

This author was supported by the Institute for Mathematics and Its Applications (IMA) and NSF DMS-0914674.

This author was supported by the Fields Institute, Duncan Chair of the University of Notre Dame; and NSF DMS-0712910.

This author was supported by the Duncan Chair of the University of Notre Dame; and NSF DMS-0712910.

This computation comes down to the construction of a homotopy $H(z,t)$ and tracking paths constructed from this homotopy. This tracking is carried out with standard numerical predictor/corrector methods. Each path may be tracked completely independently.

Continuation comes with the extreme parallelism of splitting the paths to be followed between different processors [**14**]. Because of this natural intrinsic parallelism, polynomial systems may be successfully handled that are much larger than is possible with other methods.

There is some redundant work in this computation, parallel or not:

Singular isolated solutions are approached by multiple paths.

Even worse, tracking a path leading to a singular solution is computationally much more expensive than the cost of tracking a path leading to a nonsingular isolated solution. The extra cost for computing a path leading to a singular solution is due to the necessity of using special numerical methods called endgames [**11, 12, 13, 18, 7**] needed to deal with then, and the endgames currently used do not take advantage of parallelism. For an extended discussion of the classical endgames, see [**17**, Chapter 10].

This article surveys endgames in the presence of parallelism. We introduce a novel parallel algorithm for performing endgames at the end of homotopy paths, based on the Cauchy endgame, along with some heuristics useful in its implementation. This method, which has been implemented in the Bertini software package, leads to a significant gain in efficiency. A brief description of how this is done follows.

The system we want to solve $f(z) = 0$ is embedded in a system $H(z,t)$ of N polynomials with $(z,t) \in \mathbb{C}^N \times \mathbb{C}$, such that

1. $H(z,0) = f(z)$;
2. the set of nonsingular isolated solutions $\mathcal{F}$ is given for the polynomial system $H(z,1) = 0$;
3. given any element x^* of the finite set $\mathcal{F}$ of nonsingular isolated solutions of $H(z,1) = 0$, the connected component $\mathcal{C}$ of the set
$$\{(z,t) \in \mathbb{C}^N \times (0,1] \,|H(z,t) = 0\} \tag{1.1}$$
that contains x^* is the graph of a differentiable map $t \to x(t)$ with $x(1) = x^*$; and
4. given any isolated solution $\widehat{x}$ of $f(z) = 0$ of multiplicity μ, there are exactly μ points in $\mathcal{F}$ with associated paths having the limit $\widehat{x}$ as t goes to 0.

Most of the effort in the numerical computation of the solution sets of systems of polynomials comes down to computing the limits as t goes to 0 for solution paths $x(t)$ starting at the solution $x^* \in \mathcal{F}$.

For example, if

$$f(z) = \begin{bmatrix} f_1(z_1, \ldots, z_N) \\ \vdots \\ f_N(z_1, \ldots, z_N) \end{bmatrix} \tag{1.2}$$

is a system of polynomials on $\mathbb{C}^N$, there are numerous ways to set up homotopies $H(z,t)$ with $H(z,0) = f(z)$ [**17**]. For such homotopies, the set of *startpoints* $\mathcal{F}$ consists of all isolated nonsingular solutions of the *start system* $H(z,1) = 0$. The set of *endpoints*, i.e., limits of the solution paths starting at points of $\mathcal{F}$, contains all isolated solutions of $f(z) = 0$, the *target system.* The special variable t is called the *path variable* or *parameter* for the homotopy. Although it is sometimes useful to allow for multiple parameters, we restrict to the case of a single parameter in this article. For more background, [**1, 8, 9, 17**] are good references for numerical homotopy methods.

If an isolated solution $\widehat{x}$ of $H(z,0) = 0$ is the limit of some number $d \geq 1$ solution paths $x(t)$ as $t \to 0$, then d is the *multiplicity* μ of $\widehat{x}$ as a solution of $H(z,0) = 0$. Singular isolated solutions of $f(z) = 0$, i.e., isolated solutions with multiplicity $\mu \geq 2$, are typically much more expensive to compute than nonsingular solutions. This is because the Jacobian matrix is ill-conditioned near such singular limit points, requiring the use of costly higher precision and shorter steplengths. Additionally, all of the paths leading to the same limit point will be followed since, *a priori*, there is no way to decide which paths will lead to the same solution.

To help mitigate the numerical difficulties encountered when trying to compute singular endpoints, a variety of techniques more sophisticated than standard path tracking have been developed [**7, 11, 12, 13, 17**]. One of these is the Cauchy endgame [**11**]. As described in §3, this endgame employs a numerical version of Cauchy's integral formula on loops in the parameter space around $t = 0$ to produce approximations of the endpoint at $t = 0$.

This article starts with a brief overview of endgames in §2. In §3, we discuss the classical Cauchy endgame. In §4, we present our new parallel endgame which is based on the fact that using the classical Cauchy endgame, we can identify those paths which lead to the same limit point *without* tracking every path all the way to convergence at $t = 0$. Besides reducing the work in computing the same endpoint, this significantly decreases the amount of post processing needed to decide which endpoints are the same. Not only does the Cauchy endgame have the above compelling properties, but, unlike other endgames, it parallelizes well, as demonstrated by examples in §5.

We also present a pair of heuristics §4.2 for deciding whether a given value of t is less than the modulus of all nonzero singular points in the parameter space (i.e., the nonzero ramification points). This is valuable information since Cauchy loops with images containing nonzero singular points are useless in computing approximations at $t = 0$. Skipping useless Cauchy loops (i.e, not beginning the endgame until all Cauchy loops are worth computing) clearly saves computation time.

2. Overview of endgames

Endgames are based on the uniformization theorem for germs of one-dimensional analytic sets, i.e., given a point x^* on a complex analytic curve C, there are sufficiently small open connected neighborhoods U such that $U - x^*$ is a union of punctured disks. In the situation of a homotopy $H(z,t) = 0$ on $\mathbb{C}^N \times \Delta$ that is polynomial in z and t, we have a point x^* such that $H(x^*, 0) = 0$ and at least one path leads to x^*. Let X be the union of the one-dimensional irreducible components of the solution set of $H(z,t) = 0$ that the projection map $\pi(z,t) = t$ is generically finite-to-one. Given a sufficiently small $\epsilon > 0$ and letting $\Delta(\epsilon) := \{t \in \mathbb{C} \mid |t| < \epsilon\}$, the analytic set $X \cap \pi^{-1}(\Delta(\epsilon))$ is a union on analytic sets C_i for $i = 1, \ldots, k$ such that

(1) there is a one-to-one holomorphic map $\phi : s \in \Delta\left(\epsilon^{\frac{1}{w_i}}\right) \to C_i$ with $\phi(0) = x^*$;

(2) $\pi : C_i \to \Delta(\epsilon)$ is proper and the composition $\pi \circ \phi$ maps s to t^{w_i}.

The number w_i is called the *winding number* of C_i. If x^* is an isolated solution of $f(z) = H(z, 0)$ of multiplicity μ, then $\mu = w_1 + \cdots + w_k$.

In the classical special case when $N = 1$, the fractional power series expansions of projections of function $\sigma : C_i \to \mathbb{C}$ under the map $(z, t) \to z$, are called Puiseux expansions and were classically used to investigate the local structure of curves and compute local invariants such as the multiplicity of a singularity, e.g., [**15**, Chapter V].

Let σ denote the projection $(z, t) \to z$.

The idea of the first endgame [**13**] to be developed was to send $t \in \Delta(\epsilon)$ to the sum of the points $\sigma\left(X \cap \pi^{-1}(\Delta(\epsilon))\right)$ counted with multiplicity. This function, called the *trace* of $H(z, t) = 0$ at x^*, is holomorphic and has a limit as $t \to 0$ equal to the point x^*. It has the added advantage that if in fact the curves C_i have different limits x_i^*, then the the trace converges to $\displaystyle\sum_{i=1}^{k} \frac{w_i}{\mu} x_i^*$. This robustness is balanced against the fact that we need to decide which points are going to a given x^* and then synchronize the paths going to the solution x^* so we can track the trace as $t \to 0$.

The second endgame [**11**] is the Cauchy test. This test uses the fact that $z_j \circ \phi_i : s \to \mathbb{C}$ is holomorphic and the j^{th} coordinate of the solution x^* may be computed using Cauchy's formula for $z_j \circ \phi_i(x^*)$ in terms of the integral of $z_j \circ \phi_i$ over the circle $s = c$ for sufficiently small constants c. All quantities in question may be computed in terms of continuation paths in the t-plane. This is discussed more thoroughly below.

The third endgame [**12**] is the power-series endgame, which uses interpolation theory to build up the power series expansions of the $z_j \circ \phi(s)$ and then evaluate them at $s = 0$ to find an estimate of x^*. The key point is to decide the winding number w_i. This may be done by using the same tracking as in the case of the Cauchy endgame, but this would require most of the work of the Cauchy endgame with no real improvement in the quality of the estimate for x^*. The main advantage of the power series endgame is that if w_i is not too large, the values of the coordinates of the points on the continuation paths with t real may be used to estimate the winding number.

The Cauchy and the power-series endgames require only that $H(z, t)$ is polynomial in the z variables and complex analytic in the t variable.

It is appropriate here to mention that with only double precision, solutions with winding numbers beyond 3 or 4 are hard to compute. A rough rule-of-thumb in practice is that using arithmetic with K digits and with a winding number of w_i, we can only compute the solution to K/w_i digits of accuracy. So to compute a solution x^* to four digits of accuracy with 16 digits of precision, its winding number cannot be much beyond 4. Solutions at infinity often have large winding numbers. The article [**7**] considers polyhedral methods for dealing with infinite solutions.

3. The Cauchy endgame

3.1. Background. The Cauchy endgame was first reported in [**11**] as one of several potential endgames for use with homotopy continuation (see also [**17**]). The idea is to use the Cauchy integral formula to approximate $x(0)$ using loops in the parameter space about 0 of radius t, for varying values of t. In this section, we describe the theory behind this numerical method; the next section contains a few implementation notes. Since most homotopies have $\mathbb{C}$ as the parameter space, we will assume that our parameter space is $\mathbb{C}$.

Each isolated solution x^* of $H(z, 0)$ is the endpoint of at least one solution path of the homotopy. In fact, there is some number of paths, d, with endpoint x^*. For a small value of t, say t', near enough 0, the path values of these d paths will be permuted by moving around the loop $t'e^{i\theta}$ for $\theta \in [0, 2\pi]$. In fact, the corresponding permutation can be broken up into some number of cycles, say k. This decomposition yields the winding numbers w_i of the previous section.

Given a point a in an open subset $U \subset \mathbb{C}$ and a disk D with $a \in D \subset U$, the Cauchy integral formula computes the function value $f(a)$ as

$$f(a) = \frac{1}{2\pi i} \int_C \frac{f(z)}{z - a} dz, \tag{3.1}$$

where C is the boundary of disk D and the integral is taken to be a contour integral. A numerical integration method, e.g., the trapezoid method, may be used to approximate the integral using just a sampling of points on D. Due to the periodicity, the trapezoid method is more accurate than one might have expected based on standard error estimates. Moreover, it can even be more accurate than Simpson's method [**5**, § 2.9].

In the homotopy setting, we compute approximations to a solution curve $x(t)$ of $H(z, t)$ for values of t from 1 to 0. As t approaches 0, starting at $t = 0.1$, for example, we may begin computing approximations of $x(0)$, using loops in the parameter space of the form $te^{i\theta}$ and a numerical method for evaluating the integral above. As t marches to 0, these approximations will converge, so we may terminate this endgame (and the path) once two consecutive approximations are within a prescribed tolerance.

3.2. Implementation. The Cauchy endgame has been implemented in Bertini [**2**]. This section describes the original implementation of this endgame in Bertini, though it has recently been improved using the ideas from §4.

Given a solution path $x(t)$, Bertini will first track the path to the endgame boundary, t', set to 0.1 by default. At that point, Bertini sends the path variable t around the circle $t'e^{i\theta}$ for $\theta \in [0, 2\pi]$. Since it is simpler to follow lines than curves in t, Bertini discretizes this circle into a polygon (an octagon by default) and tracks around this polygonal path repeatedly until the path value of $x(t)$ at the end of a loop is very near the starting path value $x(t')$ (within a prescribed tolerance), i.e., until the Cauchy loop closes.

Once the Cauchy loop closes, it is straightforward to use numerical integration techniques, e.g., the trapezoid method, with the Cauchy integral formula to compute an approximation at $t = 0$. After this first approximation of $x(0)$ has been computed, t' is reduced by some factor (typically $\frac{1}{2}$), and the procedure is repeated. The endgame terminates once two consecutive approximations of $x(0)$ agree to the desired tolerance.

4. A new Cauchy endgame

4.1. Overview. The motivation for the revised Cauchy endgame is to recognize those paths which coalesce to some solution x^* of $H(z, t) = 0$ at $t = 0$ without wasting the time needed to follow all paths to $t = 0$ and determine which endpoints are the same. Let $t = t^*$ be the value of the path variable from which the approximation to $t = 0$ finally converges for some given path. Let w be the winding number of x^* for the given path.

At $t = t^*$, we will have tracked around $t = 0$ w times, thereby collecting all w solutions of $H(z, t^*) = 0$ which lead to x^* for this cycle. Call this set of points $\mathcal{G}$. We may then follow the paths beginning at points of $\mathcal{G}$ backwards to discover which of the solutions of $H(z, 1) = 0$, i.e., which starting points, lead to x^*.

At first glance, it may seem that there is little savings in this method over the original Cauchy endgame. However, endgames are the expensive part of path tracking. Thus, being able to avoid running the endgame for $w - 1$ of the w paths in the same cycle leading to the same endpoint, for example, saves a significant amount of computational time, particular if $w \gg 1$. Of course, this back-tracking is unnecessary if $w = 1$.

4.2. Heuristics about when to start the endgame. The Cauchy endgame can be very inefficient when extraneous Cauchy loops are computed where convergence probably will not occur. In this section, we present a pair of heuristics that help to eliminate this wasted computational effort. After passing both heuristic tests, the standard Cauchy endgame is utilized with the goal of being in a region where convergence can be attained quickly.

Endgames are known to converge within an annulus around $t = 0$ [**17**]. The Cauchy endgame requires that no singular point other than (possibly) $t = 0$ falls in the interior of the image of the Cauchy loop. Thus, the outer loop of the annulus is determined by the (unknown) set of singular (ramification) points in the parameter space. The inner loop is determined by the preponderance of numerical error very near $t = 0$ when using fixed precision. This annulus is called the *endgame operating zone*. Since only the outer loop matters when using adaptive precision [**3, 4**], the radius of that loop is given a name, the *endgame convergence radius*. Note that the actual region from which the endgame may converge is a superset of the endgame operating zone, but, in practice, it is reasonable to think of this region as an annulus.

Let $w \geq 1$ be the winding number for the endpoint of the path $x(t) : \mathbb{C} \to \mathbb{C}^n$ at $t = 0$. In the endgame operating zone, it is known that $x(t)$ has a Puiseux series expansion, namely $x(t) = x(0) + \sum_{j=1}^{\infty} a_j t^{j/w}$. To avoid trivialities in this section, we shall assume that $x(t)$ is nonconstant. Let $c = \min\{j | a_j \neq 0\}$ and let $v \in \mathbb{C}^n$ be a random vector. Then,

$$v \cdot x(t) = v \cdot x(0) + v \cdot a_c t^{c/w} + \sum_{j=c+1}^{\infty} v \cdot a_j t^{j/w} \tag{4.1}$$

with $v \cdot a_c \neq 0$.

The first heuristic method approximates the value of $\frac{c}{w}$ and tests for agreement between two such approximations. To approximate this value, three points along the path $x(t)$ are collected in a geometric sequence, say $x(R), x(\lambda R)$, and $x(\lambda^2 R)$, for some $0 < \lambda < 1$ and $R > 0$. Using Eq. 4.1,

$$g(R) := \frac{\log \left| \frac{v \cdot x(\lambda R) - v \cdot x(\lambda^2 R)}{v \cdot x(R) - v \cdot x(\lambda R)} \right|}{\log \lambda} \approx \frac{c}{w}. \tag{4.2}$$

To pass the heuristic test, $g(R)$ must be positive, and $g(R)$ and $g(\lambda R)$ must "agree." For some $0 < L < 1$, agreement is defined as

$$L < \frac{g(R)}{g(\lambda R)} < \frac{1}{L}. \tag{4.3}$$

Bertini uses $L = \frac{3}{4}$, and, as a fail-safe mechanism to avoid the possibility of never passing the test due to numerical error, Bertini automatically moves on to the next heuristic test if it tracks to a value of t which is smaller than 10^{-8}.

The second heuristic method compares values around the Cauchy loops. When a loop contains an erroneous branch point or the radius is too large for convergence, the values around the loop generally differ by large amounts. One way of enforcing that the values do not differ radically without running the whole Cauchy endgame is to collect sample points on $x(Re^{i\theta})$, $\theta \in [0, 2\pi]$ and determine if the minimum and maximum norms of these sample points, say m and M, respectively, are "well-behaved." For $\beta > 0$ and $0 < K < 1$, m and M are well-behaved if either $M - m < \beta$ or $\frac{m}{M} > K$. Bertini takes β to be the requested final tolerance of the endpoint and $K = \frac{1}{2}$. As in the first heuristic test, Bertini employs a fail-safe mechanism to automatically start the standard Cauchy endgame if it tracks to a value of t which is smaller than 10^{-14}.

4.3. Parallel version. In the standard single-processor version of homotopy continuation, paths are tracked sequentially, i.e., one after another. One way to parallelize homotopy continuation is to send a packet of paths to each processor. Though this way of parallelizing homotopy continuation is straightforward, there is value in considering how to distribute the work among the available processors to minimize total running time.

In Bertini, the paths are dynamically distributed in packets with the size of the packets in successive rounds of distribution decreasing exponentially. That is, the size of the first packet is substantially larger than the size of the last packet. This provides for a more uniform load balance for the processors since not all paths take the same amount of time. Indeed, those ending at singular endpoints or passing near a singularity take considerably longer than those which stay well-conditioned throughout the entire path.

In the Bertini implementation of the new endgame described above, the manager maintains a list of startpoints for which the endpoint is unknown. When a worker process is available, the manager sends it a packet of startpoints, with the sizes of the packet decreasing exponentially, as before. The worker process sequentially computes the endpoint of the path for each of the start points it received and back-tracks when necessary. Before tracking each path, the startpoint is compared with the track-back points computed for this packet. After each endpoint is known for the startpoints in the packet, the data is sent back to the manager who updates the list of startpoints and sends another packet.

This parallelization can result in running the endgame more than using a sequential processing if the paths on the same cycle are simultaneously sent to different workers. The additional communication costs to avoid this would, in general, be more expensive than the cost of the extra computations. By reducing the maximum size of the packets, the likelihood of such an event occurring decreases, but this creates more communication between the manager and the workers. To maintain a good balance, we found that a maximum of 20 paths per packet works well.

5. Implementation details and computational results

The track-back Cauchy endgame is implemented in the software package Bertini [**2**]. All the examples discussed below were run on a 2.4 GHz Opteron 250 processor with 64-bit Linux. The parallel examples were run on a cluster consisting of a manager that uses one core of a Xeon 5410 processor and 8 computing nodes each containing two 2.33 GHz quad-core Xeon 5410 processors running 64-bit Linux, i.e., one manager and 64 workers.

In the examples presented, the paths were tracked using adaptive precision [**3, 4**]. The power series endgame collected sample points along the path at $t = 4^{-k}$, $k = 1, 2, \ldots$, and used four successive sample points to approximate the endpoint. Similarly, the Cauchy endgame computed approximations of the endpoint at $t = 4^{-k}$, $k = 1, 2, \ldots$ using four sample points per loop. For both endgames, the stopping criterion was having two successive approximations agree to a tolerance of 10^{-10}.

5.1. Solutions at infinity. Solutions at infinity tend to have large winding numbers, which leads to computational difficulty when trying to compute the endpoints accurately. Solving the first stage of the cascade algorithm [**16, 17**] is one place where solutions at infinity waste computational resources.

For example, on $\mathbb{C}[x, y, z]$, let B be a random 3×3 unitary matrix over $\mathbb{C}$, L_1, L_2, and L_3 be general linear functions,

$$L = \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix},$$

and

$$g(x, y, z) = \begin{bmatrix} (y - x^2)(x^2 + y^2 + z^2 - 1)(2x - 1) \\ (xy - z)(x^2 + y^2 + z^2 - 1)(2y - 1) \\ (xz - y^2)(x^2 + y^2 + z^2 - 1)(2z - 1) \end{bmatrix}.$$

The first stage of the cascade algorithm solves the polynomial system

$$f = g + BL. \tag{5.1}$$

Bertini's theorem [**17**] provides that f has only nonsingular isolated solutions on $\mathbb{C}[x, y, z]$. Using a total degree homotopy, 36 paths lead to the nonsingular isolated solutions and 89 diverge to infinity. Table 1 lists the winding numbers occurring when solving f with a total degree homotopy and the average time per path for each of the winding numbers which occur. Table 2 compares the serial version of the track-back Cauchy endgame with the classical Cauchy endgame and the power series endgame. In particular, using the track-back Cauchy endgame, the endgame had to be run on only 26 of the 89 paths that diverge, resulting in less total computation time.

5.2. A family of examples. Due to the nature of the track-back Cauchy endgame, it is more advantageous over the traditional endgame when there are many paths with large winding numbers. To illustrate this, consider a family of examples generated from [**6, 19**]. For $n \geq 3$, define

$$f_i(x_1, \ldots, x_n) = x_i^n - \prod_{j \neq i} x_j \tag{5.2}$$

winding number	number of paths	Endgame	
		Cauchy	Power Series
1	41	0.010	0.007
2	4	0.092	0.084
4	72	0.023	0.156
8	8	0.734	2.641

TABLE 1. Distribution of the winding number for solving Eq. 5.1 and average time, in seconds, for running the endgame

total paths	track-back paths removed	Endgame		
		track-back Cauchy	Cauchy	Power Series
125	63	1.63	8.31	33.89

TABLE 2. Time, in seconds, for solving Eq. 5.1 using various endgames

winding number	number of paths	Endgame	
		Cauchy	Power Series
4	4	0.015	0.004
5	1250	0.020	0.037
10	500	0.038	0.512
15	75	0.055	0.698

TABLE 3. Distribution of the winding number for solving Eq. 5.2 with $n = 5$ for the paths leading to the origin and average time, in seconds, for running the endgame

n	total paths	track-back paths removed	Endgame		
			track-back Cauchy	Cauchy	Power Series
3	27	7	0.11	0.13	0.06
4	256	102	1.46	2.77	5.70
5	3125	1523	32.66	57.86	373.29
6	46,656	25,792	760.08	1662.85	10,536.13

TABLE 4. Time, in seconds, for solving Eq. 5.2 using various endgames

for $i = 1, \ldots, n$. It can be shown that there are $(n+1)^{n-1}$ nonsingular solutions and that the origin has multiplicity $n^n - (n+1)^{n-1}$, which decomposes into various cycles. Table 3 lists the winding numbers occurring for the paths that lead to the origin for $n = 5$ and the average time per path for each of the winding numbers which occur.

Table 4 compares the other endgames with the serial version of the track-back Cauchy endgame. As n increases, there is a clear increase in the percentage of the paths that are discarded by the track-back method.

Table 5 compares the parallel version of the track-back Cauchy endgame with the classical Cauchy endgame and power series endgame. As discussed in §4.3, fewer paths were removed using the parallel version of the track-back endgame than with the serial version.

n	total paths	track-back paths removed	Endgame		
			track-back Cauchy	Cauchy	Power Series
5	3125	870	4.87	5.17	9.75
6	46,656	24,745	30.58	40.19	166.10

TABLE 5. Time, in seconds, for solving Eq. 5.2 in parallel using various endgames

6. Conclusions

This article provides a short survey of continuation, particularly endgames, in the context of parallel computation. Little thought was previously put into the parallelization of endgames, so an efficient version of one particular parallel endgame was presented. In particular, the track-back Cauchy endgame reduces the number of times the Cauchy endgame is run. Homotopies which have endpoints with large winding numbers will benefit most from this advance, as seen in the examples above.

We note that analyzing the cost of different single processor endgames and the parallel Cauchy endgame are completely open topics for future research.

References

1. E.L. Allgower and K. Georg. Numerical continuation methods, An introduction. Springer Series in Computational Mathematics, vol. 13, Springer-Verlag, Berlin, 1990.
2. D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Bertini: Software for Numerical Algebraic Geometry. Available at `http://www.nd.edu/~sommese/bertini`.
3. D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, Adaptive multiprecision path tracking. *SIAM J. Numer. Anal.*, 46:722–746, 2008.
4. D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Stepsize control for adaptive multiprecision path tracking. *Contemp. Math.*, 496, 21-31, 2009.
5. P.J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Computer Science and Applied Mathematics. Academic Press, Inc., Orlando, FL, 1984.
6. B. Dayton and Z. Zeng. Computing the multiplicity structure in solving polynomial systems. Proceedings of ISSAC 2005 (Beijing, China), 116–123, 2005.
7. B. Huber and J. Verschelde. Polyhedral end games for polynomial continuation. Numer. Algorithms, 18(1):91–108, 1998.
8. T.Y. Li. Numerical solution of polynomial systems by homotopy continuation methods. In *Handbook of Numerical Analysis, Volume XI, Special Volume: Foundations of Computational Mathematics*, F. Cucker, ed., North-Holland, 209–304, 2003.
9. A. Morgan. Solving polynomial systems using continuation for engineering and scientific problems. Prentice Hall Inc., Englewood Cliffs, NJ, 1987.
10. A.P. Morgan and A.J. Sommese. Computing all solutions to polynomial systems using homotopy continuation. *Appl. Math. Comput.*, 24(2):115–138, 1987. Errata: *Appl. Math. Comput.*, 51:209, 1992.
11. A.P. Morgan, A.J. Sommese, and C.W. Wampler. Computing singular solutions to nonlinear analytic systems. *Numerische Math.*, 58:669–684, 1991.
12. A.P. Morgan, A.J. Sommese, and C.W. Wampler. A power series method for computing singular solutions to nonlinear analytic systems. *Numerische Math.*, 63:391–409, 1992.
13. A.P. Morgan, A.J. Sommese, and C.W. Wampler. Computing singular solutions to polynomial systems. *Adv. in Appl. Math.*, 13:305–327, 1992.
14. A.P. Morgan and L.T. Watson. A globally convergent parallel algorithm for zeros of polynomial systems. *Nonlinear Analysis*, 13:1339–1350, 1989.
15. J.G. Semple and G.T. Kneebone. *Algebraic Curves*. Oxford University Press, Oxford, 1959.
16. A.J. Sommese and J. Verschelde. Numerical homotopies to compute generic points on positive dimensional algebraic sets. *J. Complexity*, 16(3):572–602, 2000.

17. A.J. Sommese and C.W. Wampler. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science.* World Scientific, Singapore, 2005.
18. M. Sosonkina, L.T. Watson and D.E. Stewart. Note on the end game in homotopy zero curve tracking. *ACM Trans. Math. Software* 22(3):281–287, 1996.
19. B. Sturmfels. *Solving Systems of Polynomial Equations.* Number 97 in CBMS Regional Conference Series in Mathematics, AMS, 2002.

DEPARTMENT OF MATHEMATICS, COLORADO STATE UNIVERSITY, FORT COLLINS, CO 80523
E-mail address: bates@math.colostate.edu

DEPARTMENT OF MATHEMATICS, TEXAS A&M UNIVERSITY, COLLEGE STATION, TX 77843
E-mail address: jhauenst@math.tamu.edu

DEPARTMENT OF APPLIED & COMPUTATIONAL MATHEMATICS & STATISTICS, UNIVERSITY OF NOTRE DAME, NOTRE DAME, IN 46556
E-mail address: sommese@nd.edu