Adv Comput Math (2021) 47: 75
https://doi.org/10.1007/s10444-021-09899-y

Mixed precision path tracking for polynomial homotopy continuation

Sascha Timme[1]

Received: 20 August 2020 / Accepted: 31 August 2021 / Published online: 29 September 2021
© The Author(s) 2021

Abstract
This article develops a new predictor-corrector algorithm for numerical path tracking in the context of polynomial homotopy continuation. In the corrector step, it uses a newly developed Newton corrector algorithm which rejects an initial guess if it is not an approximate zero. The algorithm also uses an adaptive step size control that builds on a local understanding of the region of convergence of Newton's method and the distance to the closest singularity following Telen, Van Barel, and Verschelde. To handle numerically challenging situations, the algorithm uses mixed precision arithmetic. The efficiency and robustness are demonstrated in several numerical examples.

Keywords Numerical path tracking · Adaptive step size control · Homotopy continuation · Numerical nonlinear algebra.

1 Introduction

Systems of polynomial equations arise in many applications across the sciences including computer vision [20, 36], chemistry [28], kinematics [41], and biology [32]. A numerical method for finding all isolated solutions of a system F of n polynomials in n unknowns is homotopy continuation [35]. This method constructs a homotopy $H(x,t) : \mathbb{C}^n \times \mathbb{C} \to \mathbb{C}^n$ such that $H(x,t)$ is a polynomial system for all t, $H(x,1) = F(x)$ and $H(x,0)$ is a system whose isolated solutions are known. There is a well-developed theory [35] on how to construct such homotopies to guarantee, with probability one, that every isolated solution of $F(x) = 0$ is the endpoint

Communicated by: Jon Wilkening

✉ Sascha Timme
timme@math.tu-berlin.de

[1] Chair of Discrete Mathematics/Geometry, Technische Universität Berlin, Berlin, Germany

of at least one smooth solution path $x(t)$. A solution path $x(t)$ is implicitly defined by the conditions

$$H(x(t), t) = 0 \text{ for all } t \in [0, 1) \text{ and } x(0) = x_0 \; . \tag{1}$$

In order to trace a path $x(t)$, the problem (1) is treated as a sequence of problems

$$H(x(t_k), t_k) = 0 \; , \quad k = 0, 1, 2, \ldots \tag{2}$$

with an (a priori unknown) subdivision $0 = t_0 < t_1 < \ldots < t_M = 1$ of the interval [0, 1]. Each of the problems (2) is then solved by a *correction method*, usually Newton's method, under the assumption that a *prediction method*, e.g., Euler's method, provides a good starting point. Often, the choice of step size $\Delta t_k := t_{k+1} - t_k$ is given by an *adaptive step size control*. The step size must be chosen appropriately: if the step size is too large, the prediction can be outside the zone of convergence of the corrector, while a too small step size means progress is slow. There have been many efforts to design such adaptive step size controls [17, 25, 33].

In the context of polynomial homotopy continuation methods, two phenomena need particular attention. Polynomial systems often have singular solutions, and thus, the paths leading to these solutions are necessarily ill-conditioned at the end. While endgame methods [30, 31] exist to compute singular solutions, these still require to track the solution path sufficiently close to the singularity. Usually, homotopies guarantee, with probability one, that no path passes through a singularity before reaching its endpoint. However, there is a non-negligible chance that a near-singular condition is encountered during the tracking.

Also, if two different solution paths are near to each other, then this can cause *path jumping*. That is, the solution that is tracked "jumps" from one path to another. The typical reason is that starting from a point on the tracked path the prediction method returns a point that, according to the correction method, is a numerical approximation of a point that is on a different path than the tracked one. A possible result of path jumping is that not all isolated solutions of a polynomial system are computed. Recently, Telen, Van Barel, and Verschelde [37] introduced an algorithm that is very robust against path jumping. An implementation of this algorithm is available in the software package `PHCpack` [39].

Therefore, path tracking algorithms are required to reduce the risk of path jumping and they need to be able to handle ill-conditioned situations during the tracking. Existing software packages, e.g., Bertini [6], use a version of the following path tracking algorithm. The algorithm has the following parameters: An initial step size $\Delta t > 0$, a number of corrector iterations $N \geq 1$ allowed per step, a step adjustment factor $\lambda \in (0, 1)$, a step expansion integer $E \geq 1$, and a minimum step size $t_{\min}$. Additionally, there is a tracking tolerance $\varepsilon > 0$. This means that for a given t an approximate solution $x \approx x(t)$ has to satisfy a normwise absolute error $\|x - x(t)\|_\infty \leq \varepsilon$.

Given an approximate solution $x \approx x(t)$ the prediction method provides an initial guess $\hat{x}(\Delta t) \approx x(t + \Delta t)$. Then, Newton's method iteratively improves the approximation $\hat{x}(\Delta t)$. If the required tracking tolerance ε is achieved with a most N iterations, then the solution is updated and $t = t + \Delta t$. If there are E successes in a row, then the step size is expanded to $\Delta t = \lambda^{-1} \Delta t$. If on the other hand the tolerance is not achieved with at most N iterations, then the step size is reduced to $\Delta t = \lambda \Delta t$. If $\Delta t < t_{\min}$, the algorithm terminates with a failure. Otherwise the procedure is repeated until $t = 1$ is reached.

The key to avoid path jumping is to allow only a small number of Newton iterations, typically only $N = 2$ or $N = 3$. In practice, this is often sufficient for the initial guess $\hat{x}(\Delta t)$ to stay within a small enough region surrounding the path such that no path jumping occurs. However, if two paths are closer than the required tracking tolerance ε for some $t^* \in (0, 1)$, then this algorithm tends to fail for these paths. This is shown in the computational experiments in Section 5 for two different examples. Therefore, it is necessary to choose the path tracking tolerance ε smaller than the minimal pairwise distance of any two paths. However, knowing the optimal choice of ε a-priori is impossible. Thus, one has to use either a pessimistic value for ε or resort to trial and error. But choosing ε small does not only slow down the tracking of *all* paths, it also can result in new tracking failures. The reason for this is that Newton's method in floating point arithmetic *cannot* always produce solutions whose relative normwise error is smaller than ε. This was shown by Tisseur in [38] and is explained in detail in Section 2.2.

To avoid tracking failures due to insufficient precision, Bates, Hauenstein, Sommese, and Wampler [5] developed an adaptive precision version of the above described path tracking algorithm. During the tracking, the algorithm dynamically changes the working precision such that Newton's method can theoretically always produce solutions accurate enough for the desired tracking tolerance. This eliminates the problem of insufficient precision in exchange of a possibly high computational cost. But it also still leaves open the problem of picking a suitable tolerance ε.

This article introduces a new path tracking algorithm, Algorithm 3, that does not require the choice of a path tracking tolerance ε or a maximal number N of corrector iterations allowed per step. This allows the algorithm to handle numerically challenging situations. The key idea is to use a more intrinsic measure for accepting an initial guess in the Newton corrector: An initial guess $\hat{x}(\Delta t)$ should only be accepted if the Newton iterates $\hat{x}(\Delta t) = x^{(0)}, x^{(1)}, x^{(2)}, \dots$ satisfy

$$\|x^{(j+1)} - x^{(j)}\| \leq a^{2^j - 1} \|x^{(1)} - x^{(0)}\| \tag{3}$$

for $j = 1, 2, \dots$ and some fixed constant $a \in (0, \frac{1}{2}]$. If the initial guess satisfies (3), then it is an *approximate zero*. This notion was introduced by Smale [34] for $a = \frac{1}{2}$ and plays an important role in the complexity analysis of polynomial homotopy continuation methods [11, 26]. Based on this idea, this article develops a new Newton corrector algorithm, Algorithm 1. The algorithm rejects an initial guess if (3) is not satisfied for some $j = 1, \dots, m$ where m is dynamically chosen as the maximal number of iterations for which (3) can be satisfied in fixed precision floating point arithmetic.

The proposed path tracking algorithm combines the new Newton corrector algorithm with an adaptive step size control that chooses Δt based on local geometric informations. The step size control extends an adaptive step size control developed by Deuflhard [12] and combines it with the insight of [37] to use Padé approximants as prediction methods. In particular, the algorithm builds a local understanding of the region of convergence of Newton's method and following Telen, Van Barel, and

Verschelde [37] obtains an estimate of the distance to the closest singularity. This keeps the risk of path jumping low, but the algorithm cannot guarantee that path jumping does not happen. To handle numerically challenging situations, the algorithm uses *mixed precision* arithmetic. That is, while the bulk of the computations is performed in double precision, some computations are performed, if necessary, in extended precision. This article is accompanied by a prototype implementation of the algorithm and an implementation will also be available in version 2.0 of `HomotopyContinuation.jl` [10].

This article is organized as follows. Section 2.1 reviews a Kantorovich style convergence theory of Newton's method and Section 2.2 develops a new Newton corrector algorithm, Algorithm 1, based on requirement (3). In Section 3, the use of Padé approximants as prediction method is developed. In Section 4, the results from the previous sections are used to develop an adaptive step size control. Finally, the new path tracking algorithm, Algorithm 3, is stated. The algorithm's effectiveness and ability to handle challenging paths are shown through several numerical experiments in Section 5.

2 Newton's method: theory and computational aspects

The path tracking algorithm for a path $x(t)$ consists of three main components: An adaptive step size routine that provides a step size Δt, a predictor that produces an initial guess $\hat{x}$ of $x(t+\Delta t)$, and a corrector that takes $\hat{x}$ and returns either an approximation of $x(t+\Delta t)$ or rejects $\hat{x}$. This section focuses on Newton's method as a corrector. The goal is to understand the size of the region of convergence of Newton's method as well as the behavior of Newton's method in floating point arithmetic and to translate this into a Newton corrector algorithm.

2.1 Convergence results

Let $D \subseteq \mathbb{C}^n$ an open set. Let $F : D \subseteq \mathbb{C}^n \to \mathbb{C}^n$ be an analytic function and $J_F : \mathbb{C}^n \to \mathbb{C}^{n\times n}$ its Jacobian. Consider the Newton iteration

$$\begin{aligned} J_F(x^{(j)})\Delta x^{(j)} &= F(x^{(j)}) \\ x^{(j+1)} &= x^{(j)} - \Delta x^{(j)} \end{aligned}, \quad j = 0, 1, 2, \ldots \tag{4}$$

starting at the initial guess $x^{(0)} \in D$. In 1948, Kantorovich [24] already showed sufficient conditions for the convergence of Newton's method and the existence of solutions. He also showed the uniqueness region of solutions and provided error estimates. A particular property of Newton's method is that the iterates (4) are invariant under general linear transformations of F. That is, given a start value $x^{(0)} \in D$ and $A \in \mathrm{GL}_n(\mathbb{C})$, the Newton iterates of $AF(x)$ and $F(x)$ coincide. This property is referred to as *affine covariance* [13]. In the following, an affine covariant version of a Kantorovich style convergence theorem for Newton's method is given. The statement is due to Deuflhard and Heindl [14] with error bounds from Yamamoto [42].

Theorem 1 (Newton-Kantorovich [14, 42]) *Let $F : D \subseteq \mathbb{C}^n \to \mathbb{C}^n$ be analytic. For some $x^{(0)} \in D$, assume that $J_F(x^{(0)})$ is invertible and that for all $x, y \in D$*

$$\|J_F(x^{(0)})^{-1}(J_F(x) - J_F(y))\| \leq \omega\|x - y\|,$$
$$\|\Delta x^{(0)}\| = \|J_F(x^{(0)})^{-1}F(x^{(0)})\| \leq \beta$$

and $h_0 := \omega\beta \leq \frac{1}{2}$.

Let $r^* = (1 - \sqrt{1 - 2h_0})/\omega$ and $\bar{S}(x^{(0)}, r^*) := \{x \mid \|x - x^{(0)}\| \leq r^*\} \subseteq D$. Then:

1. *The iterates (4) are well-defined, remain in $\bar{S}(x^{(0)}, r^*)$, and converge to a solution x^* of $F(x) = 0$.*
2. *The solution is unique in $S(x^{(0)}, r^{**}) \cap D$ where $r^{**} := (1 + \sqrt{1 - 2h_0})/\omega$.*

Furthermore, assume $h < \frac{1}{2}$ and define the recursive sequence $h_j := \frac{h_{j-1}^2}{2(1-h_{j-1})^2}$. Then, also the following error estimates hold.

$$\|\Delta x^{(j)}\| \leq \frac{1}{2}\omega\frac{\sqrt{1 - 2h_j}}{\sqrt{1 - 2h_0}}\|\Delta x^{(j-1)}\|^2, \quad j = 1, 2, 3, \ldots \tag{5}$$

$$\|x^{(j)} - x^*\| \leq \frac{2\|\Delta x^{(j)}\|}{1 + \sqrt{1 - 2\omega\frac{\sqrt{1-2h_j}}{\sqrt{1-2h_0}}\|\Delta x^{(j)}\|}}, \quad j = 0, 1, 2, \ldots \tag{6}$$

A drawback of the Newton-Kantorovich theorem is that it is not possible to obtain sufficient conditions for the convergence of Newton's method by only using data from the initial guess $x^{(0)}$. Instead, local information about the Lipschitz constant ω is required. The necessity of local information motivated Smale to develop his α-theory [34], which only requires data from the initial guess $x^{(0)}$ to compute sufficient conditions for the convergence of Newton's method. This point of view has valuable features for the theory of computation. In particular, it is the building block for the complexity analysis of polynomial homotopy continuation methods [7, 8, 11, 26], certified path tracking algorithms [7] and the a posteriori certification of zeros [21].

Smale [34] introduced the concept of an *approximate zero* in order to talk about initial guesses that are super-convergent with the first iteration. That is, the initial guess $x^{(0)}$ converges and the constant c on the right-hand side of

$$\|\Delta x^{(j)}\| \leq c\|\Delta x^{(j-1)}\|^2, \quad j = 1, 2, 3, \ldots$$

is not large.

Definition 1 (Approximate zero) The point $x^{(0)} \in \mathbb{C}^n$ is an approximate zero of F if the Newton iterate $x^{(j)}$ is defined for $j = 1, 2, \ldots$ and satisfies

$$\|\Delta x^{(j)}\| \leq \left(\frac{1}{2}\right)^{2^j - 1}\|\Delta x^{(0)}\|\,.$$

If $x^{(0)}$ is an approximate zero, then the true zero $x^* \in \mathbb{C}^n$ of F to which the iterates are converging is the *associated zero* of $x^{(0)}$.

Smale's α-theorem gives a sufficient condition for $x^{(0)}$ to be an approximate zero. The theorem uses

$$\gamma(F,x) \ := \ \sup_{k\geq 2} \left\| \tfrac{1}{k!} J_F(x)^{-1} D^k F(x) \right\|^{\frac{1}{k-1}} \text{ and } \beta(F,x) \ := \ \|J_F(x)^{-1} F(x)\|$$

where $D^k F$ is the tensor of order-k derivatives of F and the tensor $J_F^{-1} D^k F$ is understood as a multilinear map $A : (\mathbb{C}^n)^k \to \mathbb{C}^n$.

Theorem 2 (Smale's α-theorem [34]) *There is a naturally defined number α_0 approximately equal to 0.1307 such that if $\alpha(F, x^{(0)}) := \beta(F, x^{(0)})\,\gamma(F, x^{(0)}) < \alpha_0$, then $x^{(0)}$ is an approximate zero of F.*

It is also possible to give sufficient conditions for $x^{(0)}$ to be an approximate zero under the assumptions of the Newton-Kantorovich Theorem 1.

Lemma 1 *Using notation from Theorem 1 assume*

$$h_0 = \omega \|\Delta x^{(0)}\| \leq 2(\sqrt{4a^4 + a^2} - 2a^2) =: h(a)$$

for a parameter $0 < a < 1$. *Then, the contraction factors*

$$\Theta_j := \frac{\|\Delta x^{(j+1)}\|}{\|\Delta x^{(j)}\|} \leq a^{2^j} \quad , \; j = 0, 1, 2, \ldots \tag{7}$$

and the error bounds

$$\|\Delta x^{(j)}\| \leq a^{2^j - 1} \|\Delta x^{(0)}\| \quad , \; j = 1, 2, \ldots \tag{8}$$

are satisfied. In particular, $x^{(0)}$ is an approximate zero if $h_0 \leq \sqrt{2} - 1$.

Proof From the error estimate (5) follows

$$\begin{aligned} \Theta_j \ &\leq \ \frac{1}{2}\omega \frac{\sqrt{1-2h_j}}{\sqrt{1-2h_0}} \|\Delta x^{(j)}\| \leq \frac{1}{2}\omega \sqrt{\frac{1}{1-2h_0}} \|\Delta x^{(j)}\| \\ &= \frac{1}{2}\omega \|\Delta x^{(0)}\| \sqrt{\frac{1}{1-2h_0}} \prod_{\ell=0}^{j-1} \Theta_j = \frac{h_0}{2\sqrt{1-2h_0}} \prod_{\ell=0}^{j-1} \Theta_\ell \,. \end{aligned} \tag{9}$$

From $h_0 \leq 2(\sqrt{4a^4 + a^2} - 2a^2) < \frac{1}{2}$ follows $\frac{h_0}{2\sqrt{1-2h_0}} \leq a$ and therefore

$$\Theta_j \leq a \prod_{\ell=0}^{j-1} \Theta_\ell \leq a a^{\sum_{\ell=0}^{j-1} 2^\ell} = a^{2^j} \,.$$

Statement (8) follows from (7) by observing

$$\frac{\|\Delta x^{(j)}\|}{\|\Delta x^{(0)}\|} = \prod_{\ell=0}^{j-1} \Theta_\ell \leq a^{2^j - 1} \,.$$

□

The Newton-Kantorovich theorem and Smale's α-theorem both give sufficient conditions for an initial guess to be an approximate zero. For the former, a (local) estimate of the Lipschitz constant ω needs to be obtained and for the latter γ needs to be computed. The path tracking algorithm developed in this paper is based on the Newton-Kantorovich theorem since a (rough) estimate of ω can be computed with almost no additional cost during the Newton iteration.

A computational estimate $[\omega]$ of ω is

$$[\omega] := 2\frac{\|\Delta x^{(1)}\|}{\|\Delta x^{(0)}\|^2}\,. \tag{10}$$

This can be seen as follows. Using the error estimate (5)

$$\|\Delta x^{(1)}\| \leq \frac{1}{2}\omega\sqrt{\frac{1-2h_1}{1-2h_0}}\|\Delta x^{(0)}\|^2$$

together with the observation $\frac{1-2h_1}{1-2h_0} = (1-h_0)^{-2}$ follows

$$\|\Delta x^{(1)}\| \leq \frac{1}{2}\omega\frac{1}{1-\omega\|\Delta x^{(0)}\|}\|\Delta x^{(0)}\|^2$$

which is equivalent to

$$\frac{2\|\Delta x^{(1)}\|}{\|\Delta x^{(0)}\|^2 + 2\|\Delta x^{(0)}\|\|\Delta x^{(1)}\|} \leq \omega\,. \tag{11}$$

The computational estimate (10) is now obtained by upper bounding (11) with

$$\frac{2\|\Delta x^{(1)}\|}{\|\Delta x^{(0)}\|^2 + 2\|\Delta x^{(0)}\|\|\Delta x^{(1)}\|} \leq 2\frac{\|\Delta x^{(1)}\|}{\|\Delta x^{(0)}\|^2} = [\omega]\,.$$

2.2 Computational aspects and floating point arithmetic

After establishing the theoretical foundations of Newton's method as well as a method to obtain a computational estimate of the Lipschitz constant ω, these results are now used to guide the development of a Newton corrector algorithm. For this, the behavior of Newton's method in floating-point arithmetic has to be taken into account.

Limit accuracy The following assumes the standard model of floating point arithmetic [23, Section 2.3]

$$fl(x \text{ op } y) := (x \text{ op } y)(1+\delta), \quad |\delta| \leq u, \quad \text{op} = +, -, *, /$$

where u is the unit roundoff. In standard double precision arithmetic $u = 2^{-53} \approx 2.2\cdot 10^{-16}$. In [38], Tisseur analyzed the limit accuracy of Newton's method in floating-point arithmetic. Let $x^* \in \mathbb{C}^n$ be a zero of F with $J_F(x^*)$ non-singular, and let $x^{(0)} \in \mathbb{C}^n$ be an approximate zero of F with associated zero x^*. In floating point arithmetic, we have

$$x^{(j+1)} = x^{(j)} - (J_F(x^{(j)}) + E_j)^{-1}(F(x^{(j)}) + e_j) + \varepsilon_j$$

where

- e_j is the error made when computing the residual $F(x^{(j)})$,
- E_j is the error incurred in forming $J_F(x^{(j)})$ and solving the linear system for $\Delta x^{(j)}$,
- ε_j is the error made when adding the correction to $x^{(j)}$.

Assume that $F(x^{(j)})$ is computed in the possibly extended precision $\bar{u} \leq u$ before rounding back to working precision u and assume that there exists a function ψ depending on F, $x^{(j)}$, u and $\bar{u}$ such that

$$\|e_j\| \leq u\|F(x^{(j)})\| + \psi(F, x^{(j)}, u, \bar{u})\,.$$

Similarly, assume that the error E_j satisfies

$$\|E_j\| \leq u\phi(F, x^{(j)}, n, u)$$

for some function ϕ that reflects both the instability of the linear solver and the error made when forming $J_F(x^{(j)})$. Then, the following statement holds [38, Corollary 2.3].

Theorem 3 [38] *Let $x^{(0)}$ be an approximate zero with associated zero x^*, $x^* \neq 0$, assume that $J_F(x^*)$ is non-singular, satisfies $u\,\kappa(J_F(x^*)) \leq \frac{1}{8}$ and assume that for all j*

$$u\|J_F(x^{(j)})^{-1}\|\phi(F, x^{(j)}, n, u) \leq \frac{1}{8}\,.$$

Then, Newton's method in floating point arithmetic generates a sequence of iterates $x^{(j+1)}$ whose normwise relative error decreases until the first j for which

$$\frac{\|x^{(j+1)} - x^*\|}{\|x^*\|} \approx \frac{\|J_F(x^*)^{-1}\|}{\|x^*\|}\psi(F, x^*, u, \bar{u}) + u =: \mu(x^*, u, \bar{u})\,. \qquad (12)$$

In the following, the value $\mu(x^*, u, \bar{u})$ is referred to as the *limit accuracy*. Theorem 3 shows that the limit accuracy is influenced by three factors: the working precision u, the accuracy of the evaluation of the residual (in possibly extended precision $\bar{u}$), and the conditioning of the Jacobian. The essential consequence of this is that Newton's method *cannot* always produce solutions whose normwise relative error is on the order of the working precision u. From the error estimate (5) follows that if for a given j

$$\|\Delta x^{(j)}\| \leq \frac{\omega\|\Delta x^{(j-1)}\|^2}{2\sqrt{1-2h_0}} \leq \mu(x^*, u, \bar{u})\|x^*\|, \qquad (13)$$

then the normwise relative accuracy of $x^{(j)}$ in floating point arithmetic is only of order $\mu(x^*, u, \bar{u})$. Assume that for a given j (13) is satisfied. Then, a computational estimate $[\mu]$ of $\mu(x^*, u, \bar{u})$ can be obtained by computing $\|\Delta x^{(j)}\|/\|x^{(j+1)}\|$.

The following lemma shows that using extended precision improves the limit accuracy.

Lemma 2 *For extended precision $\bar{u} \leq u$ it holds $\mu(x^*, u, \bar{u}) \approx \mu(x^*, u, u)\frac{\bar{u}}{u} + u$.*

Proof From Section 4.3.2 in [5] follows that for a system of polynomials given as a straight line program $\psi(F, x^{(j)}, u, \bar{u})$ in Theorem 3 is a linear function in $\bar{u}$, i.e., $\psi(F, x^{(j)}, u, \bar{u}) = \frac{\bar{u}}{u}\psi(F, x^{(j)}, u, u)$. The statement then follows from (12). □

If the working precision u is standard double-precision arithmetic, then computing with extended precision can be accomplished by using double-double arithmetic. A double-double number is represented as an unevaluated sum of a leading double and a trailing double, resulting in a unit roundoff of $2^{-106} = u^2$. Bailey [2] pioneered double-double arithmetic, and implementations are nowadays available for a wide variety of programming languages and architectures.

Assume that for a fixed parameter $a \in (0, \frac{1}{2}]$ the Newton iterates starting at the initial guess $x^{(0)}$ are required to satisfy the contraction factors

$$\Theta_j = \frac{\|\Delta x^{(j+1)}\|}{\|\Delta x^{(j)}\|} \leq a^{2^j} \quad j = 0, 1, 2, \ldots .$$

If the Newton iterates are computed with precision $\bar{u} = u$, then (13) implies together with Lemma 1 that if

$$\omega\mu(x^*, u, u) > a^{2^k-1} h(a)\|x^*\|, \tag{14}$$

then there does not need to exist an initial guess $x^{(0)}$ such that the first k contraction factors are satisfied. Given a fixed k, for instance, $k = 2$, this gives a criterion when to use extended precision. Similarly, if the Newton iteration is performed with extended precision, then it is possible to use only working precision again if $\omega\mu(x^*, u, u) < a^{2^k-1}h(a)$.

The working precision u is insufficient if the combination of the error in the evaluation of the Jacobian and the instability in the linear system solver become too large. In this case, a multi-precision path tracking algorithm as [5] is necessary. However, as demonstrated in Section 5, using only double precision arithmetic for the linear system solver is sufficient for most applications. Nevertheless, even if the precision u is sufficient to achieve the limit accuracy, the analysis of Tisseur also shows that the convergence speed of Newton's method can decrease due to a too unstable linear system solver. In this case, the theoretical convergence speed may not be achieved which in turn can lead to not satisfying the required contraction factors. To circumvent this, the Newton updates are improved using mixed precision iterative refinement [22] if $\bar{u} < u$. This stabilizes the linear system solver sufficiently to achieve the theoretical convergence speed.

Stopping criteria Criteria for stopping the Newton iteration are now derived. Assume that ω and the limit accuracy $\mu = \mu(x^*, u, \bar{u})$ are known. If for any j the contraction factor

$$\Theta_j = \frac{\|\Delta x^{(j+1)}\|}{\|\Delta x^{(j)}\|} \leq a^{2^j} \tag{15}$$

is not satisfied then the iteration is stopped and the initial guess is rejected. The iteration is stopped successfully at step j if the next update would be smaller than the limit accuracy, i.e.,

$$\frac{\omega\|\Delta x^{(j-1)}\|^2}{2\sqrt{1-2h(a)}} \leq \|x^{(j)}\|\mu\,. \tag{16}$$

An additional Newton update is computed to obtain a computational estimate of μ.

Scaling Before the full Newton corrector algorithm is stated, a final point is addressed. So far, a simple rescaling of variables can change the behavior of the algorithm since ω, μ and $\|\Delta x^{(j)}\|$ are not invariant under rescaling of variables. Additionally, if $x^* = 0$ the accuracy needs to be measured with an absolute, and not a relative, normwise error. A rescaling of variables is formally the change of variables

$$y = D^{-1}x, \quad D = \mathrm{diag}(d_1, \ldots, d_n),\ d_i \in \mathbb{R}_{>0}\,.$$

With $x = (x_1, \ldots, x_n)$ and $|x_i| \neq 0$ the choice $d_i \approx |x_i|$ results in new coordinates y_i of unit order. To deal with the case $|x_i| = 0$ as well as with possible overflows in floating point arithmetic, an absolute threshold value $d_{\min} > 0$ of the form

$$d_i = \max\{|x_i|, d_{\min}\} \tag{17}$$

has to be imposed. For instance, $d_{\min} = \max(\sqrt{u}\max_i |x_i|, u)$. To not introduce rounding errors, the scaling factors D should be powers of the floating-point radix β ($\beta = 2$ in the case of IEEE-754 floating point standard arithmetic). Instead of performing the change of variables explicitly in Newton's method, the size of the Newton updates can also be measured with the *weighted error* $\|D^{-1}\Delta x^{(j)}\|$. Using the scaling factors D allows the algorithm to perform independent of the initial provided variable scaling (assuming that the initial scaling is not too extreme).

The Algorithm Finally, a new Newton corrector algorithm, Algorithm 1, is stated. The algorithm builds on the results developed in this section. The idea of the algorithm is to reject an initial guess $x^{(0)}$ if the Newton iterates $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ do not satisfy

$$\|x^{(j+1)} - x^{(j)}\| \leq a^{2^j-1}\|x^{(1)} - x^{(0)}\|$$

for $j = 1, 2, \ldots, m$ and some fixed constant $a \in (0, \frac{1}{2}]$. Here, m is decided dynamically based on (16). The rejection of an initial guess is performed using the slightly stricter criterion (15). The algorithm needs as input estimates of the limit accuracy μ and the Lipschitz constant ω and also returns updated estimates of these quantities. During the path tracking, estimates are available by using the returned estimates from the previous steps. What to do at the beginning of the tracking, if these estimates are not available, is addressed after the algorithm.

The algorithm requires estimates of the limit accuracy μ and the Lipschitz constant ω. During the path tracking, the computational estimates for both of them are available by using the computed estimates of the Newton corrector from the previous step. However, this leaves open what to do for the first step. There are two possibilities. If the start solution is the solution of a previous tracking, then computational estimates of μ and ω are already available. If this is not the case, the following heuristic, Algorithm 2, to determine values for $[\mu]$ and $[\omega]$ proved to be helpful. The idea is

Algorithm 1 Newton corrector.

```
Require: $F : \mathbb{C}^n \to \mathbb{C}^n$, $x^{(0)} \in \mathbb{C}^n$, $a \in (0, \frac{1}{2}]$, estimate $[\mu] > 0$ of the limit accuracy $\mu$,
    estimate $[\omega] > 0$ of $\omega$, evaluation precision $\bar{u} \leq u$, and positive scaling factors $D$ such
    that the coordinates of $D^{-1}x^{(0)}$ are of unit order.
Ensure: Boolean indicating whether $x^{(0)}$ was accepted, approximation $\bar{x} \in \mathbb{C}^n$ of a zero $x^*$
    of $F$, updated estimate of the (limit) accuracy $\mu$ at $x^*$, updated estimate of $\omega$, number of
    updates $j$ and last contraction factor $\Theta_{j-2}$.
procedure NEWTON($F, x^{(0)}, a, [\mu], [\omega], \bar{u}, D$)
    $j \leftarrow 0$
    while true do
        $r \leftarrow$ Evaluate $F$ at $x^{(j)}$ with precision $\bar{u}$ and round result to precision $u$
        Solve $J_F(x^{(j)})\Delta x^{(j)} = r$
        $x^{(j+1)} \leftarrow x^{(j)} - \Delta x^{(j)}$
        if $j = 1$ then
            $[\omega] \leftarrow 2\frac{\|D^{-1}\Delta x^{(1)}\|}{\|D^{-1}\Delta x^{(0)}\|^2}$                ▷ Compute $\omega$ estimate
        end if
        if $j \geq 1$ and $\frac{\|D^{-1}\Delta x^{(j)}\|}{\|D^{-1}\Delta x^{(j-1)}\|} > a^{2^{j-1}}$ then          ▷ Check sufficient contraction
            return (false, $x^{(j+1)}$, $[\mu]$, $[\omega]$, $j+1$, $\frac{\|D^{-1}\Delta x^{(j)}\|}{\|D^{-1}\Delta x^{(j-1)}\|}$)
        else if $\frac{\omega\|D^{-1}\Delta x^{(j)}\|^2}{2\sqrt{1-2h(a)}} \leq [\mu]$ then          ▷ Approaching limit accuracy
            $r \leftarrow$ Evaluate $F$ at $x^{(j+1)}$ with precision $\bar{u}$ and round result to precision $u$
            Solve $J_F(x^{(j+1)})\Delta x^{(j+1)} = r$
            $x^{(j+2)} \leftarrow x^{(j+1)} - \Delta x^{(j+1)}$
            $[\mu] \leftarrow \|D^{-1}\Delta x^{(j+2)}\|$                ▷ Update of the limit accuracy
            return (true, $x^{(j+2)}$, $[\mu]$, $[\omega]$, j+2, $\frac{\|D^{-1}\Delta x^{(j+1)}\|}{\|D^{-1}\Delta x^{(j)}\|}$)
        end if
        $j \leftarrow j + 1$
    end while
end procedure
```

to add a small perturbation to the provided start solution and to perform two Newton steps. If the perturbation is sufficiently small, then the perturbed solution still converges to the provided start solution, and an estimate of $[\omega]$ and $[\mu]$ can be obtained. As an added benefit, this provides a test to point out invalid start solutions, e.g., due to user error. For simplicity, it is assumed that it is sufficient to compute the residual with precision u.

3 Predictors and Padé approximants

After carefully studying the Newton corrector in the previous section, the attention now shifts to the predictor. Recall that the role of the predictor is to produce for a given step size Δt an initial guess $x^{(0)}$ such that the corrector converges sufficiently fast. The choice of the step size Δt will be addressed in the next section, but before it is essential to understand the influence of Δt on the distance of the initial guess $x^{(0)}$ to the solution path.

Algorithm 2 Model initialization heuristic.

```
Require: Candidate $x^{(0)} \in \mathbb{C}^n$, $a \in (0, 1)$, scaling factors $D$ such that the coordinates of
    $D^{-1}x^{(0)}$ are of unit order.
Ensure: Boolean indicating whether the initialization was successful, estimate $[\mu]$ of the limit
    accuracy $\mu$ of the associated zero of $x^{(0)}$, and an estimate $[\omega]$ of $\omega$.
procedure ModelInitialization($x^{(0)}$, $a$, $D$)
    $v \leftarrow \|D^{-1}J_F(x^{(0)})^{-1}F(x^{(0)})\| + u$
    $\varepsilon \leftarrow \sqrt{v}$
    for $k \leftarrow 1 : 3$ do                          ▷ Try up to 3 different sizes of perturbations
        $\bar{x} \leftarrow x^{(0)} + \varepsilon D$                          ▷ Add relative perturbation
        $\Delta_0 \leftarrow J_F(\bar{x})^{-1}F(\bar{x})$
        $x^{(1)} \leftarrow \bar{x} - \Delta_0$
        $\Delta_1 \leftarrow J_F(x^{(1)})^{-1}F(x^{(1)})$
        $x^{(2)} \leftarrow x^{(1)} - \Delta_1$
        if $\|D^{-1}\Delta_1\|/\|D^{-1}\Delta_0\| < a$ then
            $[\omega] \leftarrow 2\frac{D^{-1}\|\Delta x^{(1)}\|}{\|D^{-1}\Delta x^{(0)}\|^2}$
            $[\mu] \leftarrow \|D^{-1}\Delta_1\|$
            return (true, $[\mu]$, $[\omega]$)
        else                                             ▷ $\omega$ larger than $\varepsilon$, reduce $\varepsilon$
            $\varepsilon \leftarrow \varepsilon u^{2^{-k}}$
        end if
    end for
    return (false, $[\mu]$, $[\omega]$)
end procedure
```

Consider the homotopy $H(x, t) : \mathbb{C}^n \times \mathbb{C} \to \mathbb{C}^n$ and a constant $\bar{t} > 0$. Given a solution $s \in \mathbb{C}^n$ of the system $H(x, 0) = 0$ assume that there is a solution path $x(t) : [0, \bar{t}\,] \to \mathbb{C}^n$ implicitly defined by the conditions

$$H(x(t), t) = 0 \quad \text{for all } t \in [0, \bar{t}\,] \text{ and } x(0) = s\ . \tag{18}$$

Also assume that $H_x(x(t), t)$ is nonsingular for all $t \in [0, \bar{t}\,]$. Then, $x(t)$ can be extended to a holomorphic function with $H(x(t^*), t^*) = 0$ for all t^* in some nonempty open neighborhood of 0.

Without loss of generality, in the following, only the situation at $t = 0$ is considered, thus $\Delta t = t$. A predictor generates a *prediction path* $\hat{x}(t) : [0, \bar{t}\,] \to \mathbb{C}^n$ with $\hat{x}(0) = x(0)$ and they can be classified by the local order of the prediction error $\|\hat{x}(t) - x(t)\|$.

Definition 2 (Local order of a predictor) A predictor is of local order p if there exists a $\tau > 0$ and a constant $\eta_p \geq 0$ such that for all $t \in [0, \tau]$

$$\|\hat{x}(t) - x(t)\| \leq \eta_p t^p\ .$$

The constant τ is the *trust region* of the predictor.

Example 1 The Euler predictor $\hat{x}(t) = x(0) + t\dot{x}(0)$ is of local order $p = 2$ with $\tau = \bar{t}$ since

$$\|x(t) - \hat{x}(t)\| = \|x(t) - x(0) - t\dot{x}(0)\| \leq \frac{1}{2} \max_{t\in[0,\bar{t}\,]} \|\ddot{x}(t)\| t^2 .$$

There are many different families of predictors known in the literature, with the most famous ones probably being (embedded) Runge-Kutta methods. In [4] it is shown that for polynomial homotopy continuation higher-order Runge-Kutta methods are substantially more efficient than the Euler predictor. However, in the following, another particular class of predictors is considered: Padé approximants. See [3] for an exhaustive treatment of Padé approximants. The use of Padé approximants is strongly motivated by the recent results from Telen, Van Barel, and Verschelde [37] where a path tracking algorithm is developed that is very robust against path jumping. But it is to note that the use of Padé approximants as predictors was already considered in [33].

Definition 3 (Padé approximant) Let $x(t) = \sum_{\ell=0}^{\infty} c_\ell t^\ell$ be a convergent power series. The type (L, M) Padé approximant is the rational function

$$[L/M]_x := \frac{a_0 + a_1 t + a_2 t^2 + \cdots + a_L t^L}{1 + b_1 t + b_2 t^2 + \cdots + b_M t^M}$$

such that $x(t)$ and $[L/M]_x$ (considered as formal power series) satisfy

$$[L/M]_x - x(t) \in \mathcal{O}(t^{L+M+1}) .$$

Remark 1 A type (L, M) Padé approximant is a predictor of local order $L + M + 1$.

The following use of Fabry's ratio theorem is the key result from [37] which is used to obtain an estimate for the trust region of a Padé predictor. Since $x(t)$ is holomorphic in a nonempty open neighborhood of 0, there is a coordinatewise expansion of $x(t)$ as a convergent power series around 0. Write $x_j(t) = \sum_{\ell=0}^{\infty} c_\ell t^\ell$ for the Taylor expansion of the coordinate function $x_j(t)$ at 0. For sufficiently large L the pole of the Padé approximant $[L/1]_{x_j}$ indicates the *distance* to the nearest singularity (also if it is a branch point). This is seen as follows. A computation shows that if $c_L \neq 0$,

$$[L/1]_{x_j} = c_0 + c_1 t + \ldots + c_{L-1} t^{L-1} + \frac{c_L t^L}{1 - t c_{L+1}/c_L}. \tag{19}$$

Hence, the pole of $[L/1]_{x_j}$ is c_L/c_{L+1} (or it is ∞ if $c_{L+1} = 0$). Fabry's ratio theorem [16] now states that if the limit $\lim_{L\to\infty} c_L/c_{L+1}$ exists it is a singularity of $x(t)$.

Theorem 4 *Suppose that the coefficients of the power series* $\sum_{\ell=0}^{\infty} c_\ell t^\ell$ *are such that the limit* $\lim_{L\to\infty} c_L/c_{L+1} = \lambda \neq 0$ *exists. Then, the series converges uniformly inside the disk* $\{|t| < |\lambda|\}$ *and* λ *is a singular point of* $x_j(t) = \sum_{\ell=0}^{\infty} c_\ell t^\ell$.

For a fixed L the modulus $|c_L/c_{L+1}|$ therefore can be assumed to be an approximation of the distance to the nearest singularity of $x(t)$ and a computational estimate of the trust-region τ of the Padé approximant. It is recommended to see Section 3 of [37] to learn more about Padé approximants in the context of homotopy continuation.

For the computation of a Padé approximant of type (L, M), it is necessary to compute the local derivatives $x^{(\ell)}(0)$ for $\ell = 1, \ldots, L + M$. For this Mackens [27] observed the following useful identity.

Lemma 3 [27] *The local derivatives $x^{(\ell)}(t)$ can be computed using the formula*

$$x^{(\ell)}(t) = -H_x(x(t), t)^{-1} R_\ell(t)\,,$$

where

$$R_\ell(t) := \left(\tfrac{d}{d\lambda}\right)^\ell H\Big(x(t) + \textstyle\sum_{i=1}^{\ell-1} \tfrac{1}{i!} x^{(i)}(t)\lambda^i, t + \lambda\Big)\Big|_{\lambda=0}. \tag{20}$$

In [27], this identity is used for the computation of $x^{(\ell)}(0)$ by numerical differentiation. A downside of numerical differentiation is that it can suffer from catastrophic cancellation resulting in wrong approximations. Instead of using numerical differentiation, the expression (20) can be computed efficiently and accurately by using automatic differentiation [19, Chapter 13]. In particular, the cost of computing R_ℓ using automatic differentiation is at most $2\ell^2 + \mathcal{O}(\ell)$ times the cost of evaluating H by a straight-line-program. The dominating factor for the accuracy of $x^{(\ell)}(t)$ is the forward error of the linear system solving. To ensure that computed derivatives are sufficiently accurate, the forward error of the linear system solving should be monitored and if necessary be reduced by using mixed precision iterative refinement [22].

A robust Padé approximant implementation also needs to handle the edge cases that $x_j^{(\ell)}(0) = 0$ for some $1 \le j \le n$ and $1 \le \ell \le L + M$. In [18], a robust algorithm is proposed for computing Padé approximants. The provided implementation uses this algorithm for the computation of the Padé approximants.

It is also possible to obtain an estimate of the local approximation error of a Padé approximant as was shown in [37]. By comparing for each coordinate function $x_j(t)$ the coefficient of t^{L+M+1} in

$$(a_0 + a_1 t + \ldots + a_L t^L) - (1 + b_1 t + \ldots + b_M t^M)(c_0 + c_1 t + c_2 t^2 + \ldots)$$

it follows

$$e_{0,j} = -(c_{L+M+1} + b_1 c_{L+M} + \ldots + b_M c_{L+1})\,.$$

Considering the Taylor expansion of $[L/M]_{x_j}$ at 0 it follows

$$x_j(t) - [L/M]_{x_j}(t) = e_{0,j} t^{L+M+1} + \mathcal{O}(t^{L+M+2})\,.$$

Therefore, for a Padé approximant, a computational estimate $[\eta_{L+M+1}]$ of η_{L+M+1} is

$$[\eta_{L+M+1}] := \|D^{-1}(e_{0,1}, \ldots, e_{0,n})\| \tag{21}$$

where $D > 0$ are the same scaling factors as used for the Newton corrector in Section 2.

4 Step size control and path tracking algorithm

After studying Newton's method as a correction method in Section 2 and Padé approximants as a prediction method in Section 3, the results are now combined to derive an adaptive step size control. Afterward, the path tracking algorithm is stated.

Consider the homotopy $H(x,t) : \mathbb{C}^n \times \mathbb{C} \to \mathbb{C}^n$ with a given solution $s \in \mathbb{C}^n$ of the system $H(x,0) = 0$ and constant $\bar{t} > 0$. Assume there is a solution path $x(t) : [0, \bar{t}] \to \mathbb{C}^n$ implicitly defined by the conditions $x(0) = s$ and (18), and assume that $H_x(x(t),t)$ is non-singular for all $t \in [0, \bar{t}\,]$. Denote by $\hat{x}(t) : [0, \tau\,] \to \mathbb{C}^n$ the prediction path produced by the Padé approximant. As in Section 3, only the situation at $t = 0$ is considered such that $\Delta t = t$.

The goal of the step size routine is to provide a step size t such that the Newton iterates $x^{(j)}$ starting at the initial guess $\hat{x}(t) = x^{(0)}$ satisfy for $j = 0, 1, 2, \ldots$ the contraction factors

$$\Theta_j = \frac{\|\Delta x^{(j+1)}\|}{\|\Delta x^{(j)}\|} \leq a^{2^j} \tag{22}$$

for a fixed parameter $a \in (0,1)$, for instance $a = 0.2$. Recall from Lemma 1 that if $a \leq \frac{1}{2}$, then $\hat{x}(t)$ is an approximate zero. Assuming knowledge about the Lipschitz constant ω in a neighborhood of the path $x(t)$ and the theoretical quantities introduced in Section 3, it is possible to give a maximal theoretical feasible step size $t_{\max}$ such that this is the case. The approach to use the theoretical quantities ω, η_p, and τ to determine a maximal feasible step size such that Newton's method converges was pioneered by Deuflhard in [12].

Theorem 5 [14] *Let $D \subseteq \mathbb{C}^n$ such that for all $x \in D$ and $t \in [0, \bar{t}\,]$, $\bar{t} > 0$, the Jacobian $H_x(x,t)$ is non-singular. Assume that for each $t \in [0, \bar{t}\,]$ there exists a convex subset $D(t) \subseteq D$ with $x(t) \in D(t)$ where $x(t)$ denotes the unique solution path in $D \times [0, \bar{t}\,]$. Let $\hat{x}(t) : [0, \bar{t}\,] \to D$ denote a prediction path of order p with trust-region τ, i.e., with*

$$\|\hat{x}(t) - x(t)\| \leq \eta_p t^p \ \textit{ for all } t \in [0, \tau]\,.$$

Moreover, assume for all $t \in [0, \bar{t}\,]$ the affine covariant Lipschitz condition

$$\|H_x(\hat{x}(t),t)^{-1}(H_x(u,t) - H_x(v,t))\| \leq \omega \|u - v\| \textit{ for all } u, v \in D(t)\,.$$

For fixed $h \leq \frac{1}{2}$ let $t_{\max} = \min(t^*, \tau, \bar{t})$ where

$$t^* := \left(\frac{\sqrt{1+2h} - 1}{\omega \eta_p} \right)^{1/p} \tag{23}$$

and for all $t \leq t_{\max}$ let $B(t)$ denote a ball around $\hat{x}(t)$ with radius $(1 - \sqrt{1-2h})/\omega$ and assume $B(t) \subseteq D(t)$. Then, for all step sizes $t \leq t_{\max}$, the Newton iterates starting at $\hat{x}(t) = x^{(0)}$ are well-defined, remain in $B(t)$, converge towards $x(t)$, and satisfy $\|H_x(\hat{x}(t),t)^{-1} H(\hat{x}(t),t)\| \leq \frac{h}{\omega}$.

Proof For $h = \frac{1}{2}$ the statement is Theorem 1.3 in [12]. The more general maximal step size (23) and the inequality $\|H_x(\hat{x}(t), t)^{-1} H(\hat{x}(t), t)\| \leq \frac{h}{\omega}$ follows from equations (1.14a) and (1.14b) in the proof of Theorem 1.3 in [12]. □

In Theorem 5, there is a choice of the parameter $h \leq \frac{1}{2}$. If this is sufficiently small, then the contraction factors (22) are satisfied. The following corollary makes this precise.

Corollary 1 *In Theorem 5 choose $h \leq h(a) = 2(\sqrt{4a^4 + a^2} - 2a^2)$ for $a \in (0, 1)$. Then, the Newton iterates starting at $\hat{x}(t)$ satisfy the contraction factors*

$$\frac{\|\Delta x^{(j+1)}\|}{\|\Delta x^{(j)}\|} \leq a^{2^j} .$$

Proof Since $\|H_x(\hat{x}(t), t)^{-1} H(\hat{x}(t), t)\| \leq \frac{h}{\omega}$ follows $h_0 \leq h$ and using Lemma 1 follows the statement. □

If the step size is chosen according to Theorem 5, then path jumping cannot happen. However, to obtain the theoretical quantities ω, η_p, and τ is very hard. Instead, the theoretical quantities are replaced by easy to obtain computational estimates $[\omega]$, $[\eta_p]$ and $[\tau]$. Using the computational estimates and Corollary 1, an estimate $[t_{\max}]$ of the maximal feasible step size $t_{\max}$ is given by

$$[t_{\max}] := \min\left([t^*],\ \beta_\tau[\tau],\ \bar{t}\right) \tag{24}$$

where

$$[t^*] := \left(\frac{\sqrt{1 + 2h(a)} - 1}{\beta_\omega[\omega][\eta_p]}\right)^{1/p}$$

where $0 < \beta_\tau < 1$ and $\beta_\omega \geq 1$ are additional safety factors, for instance $\beta_\tau = 0.75$ and $\beta_\omega = 10$. Instead of choosing β_ω fixed, it seems worthwhile to develop a more adaptive criterion for choosing β_ω in the future.

Since $[\eta_p]$ and $[\omega]$ are only lower bounds and $[\tau]$ is only an upper bound for the theoretical quantities, it is possible that the step size $t = [t_{\max}]$ is larger than $t_{\max}$. Then, it can happen that the Newton corrector Algorithm 1 rejects the initial guess since $\Theta_k > a^{2^k}$ for some k. In this case, a suitable step size correction formula is

$$t' := \left(\frac{\sqrt{1 + 2h(\frac{1}{2}a)} - 1}{\sqrt{1 + 2h\left(\Theta_k^{2^{-k}}\right)} - 1}\right)^{1/p} t \tag{25}$$

which is a clear reduction since $\Theta_k^{2^{-k}} > a$.

Before the path tracking algorithm is stated, the similarities and differences to the step size control developed in [37] are stated. In [37] the authors develop an adaptive step size control similar to (24) with the difference that their algorithm uses instead of $[t^*]$ the step size candidate Δt_1 computed as follows. For Δt_1, an estimate δ of the distance to the nearest path is computed based on a second-order Taylor expansion around $x(0)$. This involves the computation of the Hessian of H and multiple singular value decompositions. Then, $\Delta t_1 = \left(\beta_1 \delta / [\eta_p]\right)^{1/p}$ where β_1 is a safety factor to unknown region of convergence of Newton's method, for instance $\beta_1 = 0.005$.

Finally, the path tracking algorithm, Algorithm 3, is stated. It is assumed that the computational estimates $[\omega]$ and $[\mu]$ for the start solution s are available. These are either available as a result of a previous path tracking or by using Algorithm 2.

5 Computational experiments

In this section, numerical experiments are shown to illustrate the effectiveness of the proposed path tracking algorithm. A prototype implementation of the path tracking algorithm together with all the data necessary to run the experiments is available at https://doi.org/10.5281/zenodo.3667414 .

The path tracking algorithm will also be implemented in version 2.0 of the Julia package `HomotopyContinuation.jl` [10].

In the following experiments, the performance of the proposed path tracking algorithm is studied. Additionally, the algorithm is compared against the adaptive precision path tracking algorithm [5] as it is implemented in the state of the art package `Bertini` [6]. For the different solvers, the following notations are used in the experiments:

`HC.jl`	The provided implementation of the algorithm
`Bertini DP`	Bertini v1.6 using double precision arithmetic (MPTYPE = 0)
`Bertini AP`	Bertini v1.6 using adaptive precision (MPTYPE = 2)

All solvers are used intentionally with the default settings unless otherwise mentioned since for a non-expert user it is very hard to understand which parameters need to be changed. Note that Bertini uses by default a path tracking tolerance of 10^{-5} before and 10^{-6} during the endgame. The experiments are performed on a 24 GB RAM machine with an Intel Core i5-7500 CPU working at 3.40 GHz. All solvers use only one core for all the experiments unless stated otherwise. The experiments are designed such that the tracked solution paths are all smooth. Therefore, endgame algorithms are not necessary. In all experiments, the implementation of the proposed algorithm uses a (2, 1) Padé approximant with parameters $a = 0.2$, $\beta_\omega = 10$ and $\beta_\tau = 0.75$.

5.1 Mixtures of Gaussians and the method of moments

This example illustrates the behavior of the path tracking algorithm with respect to the two computed quantities $[t^*]$ and $\beta_\tau[\tau]$ which restrict the step size $[t_{\max}]$ in (24). Consider two univariate Gaussian random variables X_1 and X_2 with means μ_1, μ_2 and variances σ_1^2, σ_2^2. A mixture of the two random variables X_1 and X_2 is then the random variable where for a given value $\lambda \in [0, 1]$ a value is drawn from X_1 with probability λ and with probability $1 - \lambda$ from X_2. Only given observations of the the mixture of the two random variables, the parameters $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$, and λ can

Algorithm 3 Path tracking algorithm.

```
Require: Homotopy $H(x,t) : \mathbb{C}^n \times \mathbb{C} \to \mathbb{C}^n$, $s \in \mathbb{C}^n$ such that $s$ is an approximate
    zero of $G(x) := H(x,0)$, estimates $[\omega]$ and $[\mu]$ of $\omega$ and $\mu$, $a \in (0, \frac{1}{2}]$, safety
    factors $\beta_\omega$ and $\beta_\tau$, minimal step size $t_{\min}$, type of Padé approximant $L$.
Ensure: Approximate zero of $F(x) := H(x,1)$ or false if the tracking failed.
 1: procedure TRACK($H, s, [\omega], [\mu], a, \beta_\omega, \beta_\tau, t_{\min}, L$)
 2:     $(t, \Delta t) \leftarrow (0, \infty)$
 3:     $x \leftarrow s$
 4:     $\bar{u} \leftarrow u$
 5:     Initialize scaling factors $D$ using $x$ and (17)
 6:     while $t < 1$ do
 7:         Compute $x^{(1)}(t), \ldots, x^{(L+2)}(t)$ from $(x,t)$ by using the identity (20)
 8:         Compute $[\eta_{L+2}]$ and $[\tau]$ for $(L,1)$ Padé approximant using (19) and (21)
 9:         $\Delta t \leftarrow \min\left(\left[\frac{\sqrt{1+2h(a)}-1}{\beta_\omega[\omega][\eta_{L+2}]}\right]^{1/(L+2)}, 1-t, \beta_\tau[\tau]\right)$
10:         Use $(L,1)$ Padé approximant to obtain initial guess $\hat{x}$ at $t + \Delta t$
11:         Update scaling factors $D$ using $\hat{x}$ and (17)
12:         $(\texttt{success}, \bar{x}, [\mu], [\omega], \Theta_k) \leftarrow$ NEWTON($\hat{x}, a, [\mu], [\omega], \bar{u}, D$)
13:         if success then
14:             $t \leftarrow t + \Delta t$
15:             $x \leftarrow \bar{x}$
16:         else
17:             $\Delta t \leftarrow \left(\frac{\sqrt{1+2h(0.5a)}-1}{\sqrt{1+2h\left(\Theta_k^{2^{-k}}\right)}-1}\right)^{1/(L+2)} \Delta t$
18:             if $\Delta t < t_{\min}$ then
19:                 return false
20:             end if
21:             go to Line 10
22:         end if
23:         if $\bar{u} = u$ and $[\omega][\mu] > a^5 h(a)$ then    ▷ Use extended precision, see (14)
24:             $\bar{u} \leftarrow u^2$
25:         else if $\bar{u} = u^2$ then
26:             Compute estimate $\mu_u$ of $\mu(x,u,u)$ as described in Section 2.2
27:             if $[\omega]\mu_u < a^7 h(a)$ then
28:                 $\bar{u} \leftarrow u$
29:             end if
30:         end if
31:     end while
32:     return $x$
33: end procedure
```

be recovered by using the method of moments. See [1] for a detailed algebraic treatment of this problem. As shown in [1] for a mixture of two univariate Gaussians, the method of moments results in a polynomial system consisting of five polynomials of degree two to six in five variables and with five parameters, the sample moments. For generic sample moments, this system has 18 isolated solutions. Starting from generic sample moments, the 18 solutions were tracked to 50 different real sample momements draw elementwise independently from a normal distribution. For this, a path needed on average 14.55 steps (including rejected steps) and on average only 0.18 steps got rejected. This shows that the step size control effectively avoids taking too large steps. In 62.25% of the steps, the step size was more restricted by the curvature condition $[t^*]$ than the distance to the closest singularity $\beta_\tau[\tau]$. This shows that both conditions, $[t^*]$ and $\beta_\tau[\tau]$, have also in practice an effect on the chosen step size.

5.2 Alt's problem

Alt's problem, formulated in 1923, is to count the number of four-bar linkages whose coupler curve interpolates nine general points in the plane. In 1992, Morgan, Sommese, and Wampler [40] provided a numerical proof to Alt's problem that there are generically 1442 non-degenerate four-bar linkages. Due to Roberts cognates and a twofold symmetry, the resulting polynomial system generically has 8652 regular solutions. Here the formulation of the polynomial system as an affine polynomial system in 24 variables and the 16 parameters $(\delta, \hat{\delta}) \in \mathbb{C}^8 \times \mathbb{C}^8$ is considered. Since the problem is formulated in isotropic coordinates, the physically meaningful configurations correspond to choices of parameters such that δ and $\hat{\delta}$ are complex conjugates. Consider the 'general' situation where solutions are tracked from generic parameter values $\delta_1 \in \mathbb{C}^8 \times \mathbb{C}^8$ to generic physically meaningful parameter values $(\delta_0, \bar{\delta_0})$ with $\delta_0 \in \mathbb{C}^8$. The results in Table 1 show that even this general situation results in numerically challenging paths that `Bertini AP` cannot handle with its default settings[1]. After decreasing the path tracking tolerance to 10^{-8}, in half of the cases all solutions are found. The proposed algorithm, on the other hand, reliably computes all 1442 solutions without any path jumping in a fraction of the time.

Figure 1 illustrates the behavior of the proposed algorithm for one particular numerically challenging path. This path closely passes at around $t = 0.93$ a singularity. As expected, this increases the Lipschitz constant ω and the limit accuracy μ. The limit accuracy decreases sharply as soon as the algorithm switches to extended precision. After passing the problematic region, the algorithm quickly switches back to only using double precision arithmetic, as indicated by a sharp increase of μ. The lower part of Fig. 1 depicts different values associated with the Jacobian $J = H_x(x(t), t)$ along the path: the condition number $\kappa(J)$, the componentwise relative condition number $\mathrm{cond}(J)$ and $\mathrm{cond}(J, x)$. See [23, Sec. 7] for the definitions. The

[1] It is to note that `Bertini` provides a warning when paths fail, so that a user always knows whether a provided solution set is possibly not complete.

Table 1 Results for 10 runs of Alt's problem using the same generic start solutions to a generic physically meaningful configuration

		Runtime (seconds)				# solutions		
	`tol`	Mean	Median	Min	Max	Median	Min	Max
`HC.jl`	-	6.38	6.41	5.08	7.12	1442	1442	1442
`Bertini AP`	1e-5	1931.63	1832.59	1120.43	2695.13	1436	1433	1441
`Bertini AP`	1e-8	10950.09	11218.97	8371.76	12115.66	1441	1437	1442

The `tol` column refers to the assigned path tracking tolerance. Missing solutions are a result of path tracking failures

values of $\kappa(J)$ and $\operatorname{cond}(J)$ obtain a maximum of 2.1×10^{25} resp. 1.6×10^{19}. From these values, it seems hopeless to track the path in double precision arithmetic due to the general expectation to lose around $\log_{10}(\kappa(J))$ digits of accuracy in the linear system solving. However, the componentwise relative forward error of the computed Newton updates is only governed by the much tamer $\operatorname{cond}(J, x)$ which is at most 5.8×10^{10}. This explains why it is still possible to track the path by only using mixed precision iterative refinement. Using the proposed path tracking algorithm, the path needs 253 steps in total with only two rejected steps and a total runtime of 13 ms. Trying to compute the path with `Bertini AP` results in a path failure after around 90 s and over 5000 steps due to supposedly insufficient precision of at most 1024 bits. After changing the required tracking tolerance to 10^{-12} `Bertini AP` successfully tracks the path in 120 s.

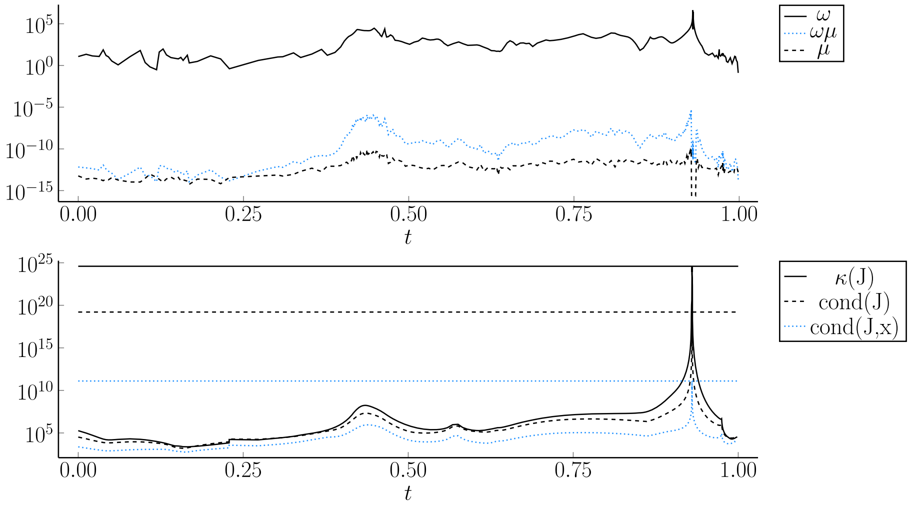

Fig. 1 Behavior of the path tracking algorithm along a single challenging path

Table 2 Results for 50 runs of Steiner's problem from generic complex parameter values to generic real parameter values

		Runtime (seconds)				# Solutions		
	tol	Mean	Median	Min	Max	Median	Min	Max
HC.jl		2.47	2.49	1.64	3.11	3264	3264	3264
Bertini DP	1e-5	34.42	34.57	23.85	45.34	3189	3094	3216
Bertini AP	1e-5	130.53	126.61	78.70	251.01	3261	3256	3264
Bertini AP	1e-8	688.50	691.59	368.36	1125.72	3264	3261	3264

The tol column refers to the assigned path tracking tolerance. Missing solutions are a result of path tracking failures

5.3 Steiner's conic problem

A classic problem in enumerative geometry is Steiner's conic problem. It asks: How many plane conics are tangent to five given conics in general position? The answer is 3264. See [15] for historic remarks and a modern intersection theory treatment of this classic problem. Here, the formulation of Steiner's conic problem from [9] is used where the resulting polynomial system consisting of 10 quadratic and 5 cubic polynomials has 15 variables and 30 parameters. To test the path tracking algorithm, consider the case of a parameter homotopy [29] from generic complex parameters $c \in \mathbb{C}^{30}$ to generic real parameters $r \in \mathbb{R}^{30}$.

The results for 50 parameter homotopies are shown in Table 2. As for Alt's problem, the proposed path tracking algorithm handles all instances without any failure or path jumping. However, even these generic instances pose problems for other path tracking algorithms with Bertini AP losing solutions almost always using the default settings. After the path tracking tolerance is manually decreased to 10^{-8} in almost all instances, all solutions are found.

6 Conclusion and future work

This article proposed a mixed precision path tracking algorithm for numerical path tracking in polynomial homotopy continuation. The results of the computational experiments demonstrate that the proposed algorithm can handle numerically challenging situations that are typical for applied problems. The implementation of the proposed algorithm significantly outperforms the available implementation of the adaptive precision algorithm [5] in the the considered examples. However, this does not eliminate the need for an adaptive precision algorithm to solve more numerically challenging problems. It is expected that the techniques in this article can be extended to develop a new adaptive precision algorithm. An implementation is available, and the algorithm will also be integrated into version 2.0 of HomotopyContinuation.jl.

Acknowledgements The author wants to thank Paul Breiding, Peter Deuflhard, Michael Joswig, Simon Telen, and Marc Van Barel for helpful discussions.

Funding Open Access funding enabled and organized by Projekt DEAL. The author was supported by the Deutsche Forschungsgemeinschaft (German Research Foundation) Graduiertenkolleg *Facets of Complexity* (GRK 2434).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Améndola, C., Faugère, J.C., Sturmfels, B.: Moment varieties of gaussian mixtures. J. Algebraic Stat. **7**(1) (2016)
2. Bailey, D.H.: A Fortran-90 based multiprecision system. ACM Trans. Math. Softw. (TOMS) **21**(4), 379–387 (1995)
3. Baker, G.A., Graves-Morris, P.: Padé approximants, 2nd edn. Encyclopedia of Mathematics and its Applications Cambridge University Press. https://doi.org/10.1017/CBO9780511530074 (1996)
4. Bates, D.J., Hauenstein, J.D., Sommese, A.J.: Efficient path tracking methods. Numer. Algorithm. **58**(4), 451–459. https://doi.org/10.1007/s11075-011-9463-8 (2011)
5. Bates, D.J., Hauenstein, J.D., Sommese, A.J., Wampler, C.W.: Adaptive multiprecision path tracking. SIAM J. Numer. Anal. **46**(2), 722–746 (2008)
6. Bates, D.J., Hauenstein, J.D., Sommese, A.J., Wampler, C.W.: Bertini: Software for numerical algebraic geometry. Available at bertini.nd.edu with permanent. https://doi.org/10.7274/R0H41PB5 (2020)
7. Beltrán, C., Leykin, A.: Robust certified numerical homotopy tracking. Found. Comput. Math. **13**(2), 253–295 (2013)
8. Beltrán, C., Pardo, L.: Smale's 17th problem: average polynomial time to compute affine and projective solutions. J. Am. Math. Soc. **22**(2), 363–385 (2009)
9. Breiding, P., Sturmfels, B., Timme, S.: 3264 Conics in a Second. Not. Am. Math. Soc. **67**, 30–37 (2020)
10. Breiding, P., Timme, S.: HomotopyContinuation.Jl: A Package for Homotopy 1Ontinuation in Julia. In: International Congress on Mathematical Software, pp. 458–465. Springer (2018)
11. Bürgisser, P., Cucker, F.: On a problem posed by Steve Smale. Ann. Math., 1785–1836 (2011)
12. Deuflhard, P.: A stepsize control for continuation methods and its special application to multiple shooting techniques. Numer. Math. **33**(2), 115–146 (1979)
13. Deuflhard, P.: Newton methods for nonlinear problems: affine invariance and adaptive algorithms, vol. 35. Springer Science & Business Media (2011)
14. Deuflhard, P., Heindl, G.: Affine invariant convergence theorems for Newton's method and extensions to related methods. SIAM J. Numer. Anal. **16**(1), 1–10 (1979)
15. Eisenbud, D., Harris, J.: 3264 and All That: A Second Course in Algebraic Geometry. Cambridge University Press (2016)
16. Fabry, E.: Sur Les Points Singuliers D'une Fonction Donnée Par Son DéVeloppement En SéRie Et L'impossibilité Du Prolongement Analytique Dans Des Cas TréS GéNéRaux. In: Annales Scientifiques De L'École Normale Supérieure, vol. 13, pp. 367–399. Elsevier (1896)
17. Gervais, J.J., Sadiky, H.: A continuation method based on a high order predictor and an adaptive steplength control. Z. Angewandte Math. Mech. **84**(8), 551–563 (2004)
18. Gonnet, P., Guttel, S., Trefethen, L.N.: Robust padé approximation via svd. SIAM Rev. **55**(1), 101–117 (2013)

19. Griewank, A., Walther, A.: Evaluating derivatives: principles and techniques of algorithmic differentiation, vol. 105. Siam (2008)
20. Hartley, R.I., Sturm, P.: Triangulation. Comput. Vis. Image Understand. **68**(2), 146–157 (1997)
21. Hauenstein, J.D., Sottile, F.: Algorithm 921: alphaCertified: Certifying Solutions to Polynomial Systems. ACM Trans. Math. Softw. **38**(4). https://doi.org/10.1145/2331130.2331136 (2012)
22. Higham, N.J.: Iterative refinement for linear systems and LAPACK. IMA J. Numer. Anal. **17**(4), 495–509. https://doi.org/10.1093/imanum/17.4.495 (1997)
23. Higham, N.J.: Accuracy and stability of numerical algorithms, vol. 80. Siam (2002)
24. Kantorovich, L.V.: On Newton's method for functional equations. pp. 1237–1240. Dokl. Akad Nauk SSSR (1948)
25. Kearfott, R.B., Xing, Z.: An interval step control for continuation methods. SIAM J. Numer. Anal. **31**(3), 892–914 (1994)
26. Lairez, P.: A deterministic algorithm to compute approximate roots of polynomial systems in polynomial average time. Found. Comput. Math. **17**(5), 1265–1292. https://doi.org/10.1007/s10208-016-9319-7 (2017)
27. Mackens, W.: Numerical differentiation of implicitly defined space curves. Computing **41**(3), 237–260 (1989)
28. Morgan, A.P.: Solving polynominal systems using continuation for engineering and scientific problems. Prentice-Hall (1987)
29. Morgan, A.P., Sommese, A.J.: Coefficient-parameter polynomial continuation. Appl. Math. Comput. **29**(2, part II), 123–160. https://doi.org/10.1016/0096-3003(89)90099-4 (1989)
30. Morgan, A.P., Sommese, A.J., Wampler, C.W.: Computing singular solutions to nonlinear analytic systems. Numer. Math. **58**(1), 669–684 (1990)
31. Morgan, A.P., Sommese, A.J., Wampler, C.W.: A power series method for computing singular solutions to analytic systems. Numer. Math. **63**(1), 391–409 (1992)
32. Narula, J., Tiwari, A., Igoshin, O.A.: Role of autoregulation and relative synthesis of operon partners in alternative sigma factor networks. PLOS Comput. Biol. **12**(12), 1–25. https://doi.org/10.1371/journal.pcbi.1005267 (2016)
33. Schwetlick, H., Cleve, J.: Higher order predictors and adaptive steplength control in path following algorithms. SIAM J. Numer. Anal. **24**(6), 1382–1393 (1987)
34. Smale, S.: Newton's method estimates from data at one point. In: Ewing, R.E., Gross, K.I., Martin, C.F. (eds.) The Merging of Disciplines: New Directions in Pure, Applied, and Computational Mathematics, pp. 185–196. Springer (1986)
35. Sommese, A.J., Wampler, C.W.: The numerical solution of systems of polynomials arising in engineering and science. World Scientific. https://doi.org/10.1142/5763 (2005)
36. Stewenius, H., Schaffalitzky, F., Nister, D.: How hard is 3-view triangulation really? In: 2005. ICCV 2005. Tenth IEEE International Conference On Computer Vision, vol. 1, pp. 686–693. IEEE (2005)
37. Telen, S., Van Barel, M., Verschelde, J.: A robust numerical path tracking algorithm for polynomial homotopy continuation. arXiv:1909.04984 (2019)
38. Tisseur, F.: Newton's method in floating point arithmetic and iterative refinement of generalized eigenvalue problems. SIAM J. Matrix Anal. Appl. **22**(4), 1038–1057 (2001)
39. Verschelde, J.: Algorithm 795: PHCpack: a general-purpose solver for polynomial systems by homotopy continuation. ACM Trans. Math. Softw. **25**(2), 251–276. http://dblp.uni-trier.de/db/journals/toms/toms25.html#Verschelde99 (1999)
40. Wampler, C.W., Morgan, A.P., Sommese, A.J.: Complete solution of the nine-point path synthesis problem for four-bar linkages. J. Mech. Des. **114**(1), 153–159. https://doi.org/10.1115/1.2916909 (1992)
41. Wampler, C.W., Sommese, A.J.: Numerical algebraic geometry and algebraic kinematics. Acta Numer. **20**, 469–567 (2011). https://doi.org/10.1017/S0962492911000067
42. Yamamoto, T.: A unified derivation of several error bounds for Newton's process. J. Comput. Appl. Math. **12-13**, 179–191. . http://www.sciencedirect.com/science/article/pii/0377042785900159 (1985)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.