Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Adaptive Step Size Control for Polynomial Homotopy Continuation Methods

by

Sascha Timme

Preprint no.: 23 2019

ADAPTIVE STEP SIZE CONTROL FOR POLYNOMIAL HOMOTOPY CONTINUATION METHODS

SASCHA TIMME

Abstract. In this paper we develop an adaptive step size control for the numerical tracking of implicitly defined paths in the context of polynomial homotopy continuation methods. We focus on the case where the paths are tracked using a predictor-corrector scheme with only a prescribed maximal number of allowed correction steps. The adaptive step size control changes the step size based on computational estimates of local geometric information, in particular a local Lipschitz constant and the local error of the used predictor method, as well as its order. The developed adaptive step size control is implemented in the software package HomotopyContinuation.jl and its efficiency over the currently commonly used adaptive step size control is demonstrated on several examples.

1. Introduction

Systems of polynomial equations arise in many applications including computer vision [HS97, SSN05], chemistry [Mor87], kinematics [WS11] and biology [NTI16]. Homotopy continuation [SW05] is a method to find all isolated complex zeros of a system $F(x)$, where $F = (f_1, \dots, f_n)$ and the f_i are polynomials in n variables. The idea is to construct a homotopy $H(x,t), H : \mathbb{C}^n \times [0,1] \to \mathbb{C}^m$ such that $H(x,1) = F(x)$ is the target system to be solved, and $H(x,0)$ is a starting system where isolated solutions are known. There is a well-developed theory on how to construct such homotopies to guarantee, with probability one, that every isolated solution of $F(x) = 0$ is the endpoint in the limit $t \to 1$ of at least one smooth path $x(t)$, implicitly defined by the conditions

$$H(x(t),t) = 0 \quad \text{for all } t \in [0,1] \text{ and } x(0) = x_0\ . \tag{1.1}$$

Equivalently, each path $x(t)$ is the solution of the initial value problem given by the *Davidenko differential equation*

$$H_x(x(t),t)\dot{x}(t) + H_t(x(t),t) = 0 \tag{1.2}$$

and the initial value $x(0) = x_0$ where H_x denotes the partial derivative of H with respect to the first argument and H_t with respect to the second one.

In order to compute a solution path $x(t)$ the problem (1.1) is treated as a sequence of problems

$$H(x(t_k),t_k) = 0\ , \quad k = 0,1,\dots \tag{1.3}$$

with a subdivision $0 = t_0 < t_1 < \ldots < t_M = 1$ of the interval $[0,1]$. Each of the problems (1.3) is then solved by a *correction method*, e.g., Newton's method, under the assumption that good starting points are provided. For each k the starting point is provided by a *prediction method* using the previous (approximate) solutions at $t_{k-1}, t_{k-2}, \dots$. Prediction

methods can be derived by discretizing the Davidenko differential equation (1.2). Due to the combination of a corrector and a predictor this scheme is referred to as the *predictor-corrector* method. In the setting of polynomial systems, the homotopies can be designed such that the paths advance monotonically, that is, there are no turning points.

For good results, the predictor step size must be chosen appropriately. A step size which is too large may result in a prediction outside the zone of convergence of the corrector, while a step size, which is too small, means progress is slow and computationally costly. Although the adaptive control of the step size is crucial for efficient path tracking, the most commonly employed step size control is rather simple: given an initial step size, the step size is decreased if a step got rejected and it is increased after a certain number of consecutive steps got accepted [BHSW08].

A phenomenon that also may occur during path tracking is *path jumping* where the corrector converges to a solution path different to where the predictor started from. While this is in some situations possible to detect, it is not possible in general. Therefore the path tracking algorithm should employ a heuristic to minimize the chance of path jumping. Prevention from path jumping can also be made robust, but unfortunately with computational cost too high for most applications [BL13].

Organization of the paper. The paper is organized as follows. In Section 2, we review a standard affine covariant convergence proof of Newton's method. This leads, in Section 3, to a novel adaptive step size control. In Section 4 we address the fact that the adaptive step size control from Section 3 is stated in affine space whereas homotopy continuation methods usually operate in projective space. This new adaptive step size control has been implemented in the software package HomotopyContinuation.jl [BT18] currently under active development by Paul Breiding and the author. Several examples are presented in Section 5 to illustrate the effectiveness of the new adaptive step size control.

2. Newton's method revisited

Due to the fundamental importance of Newton's method as a corrector in the path tracking algorithm, we recall some of its basic properties. In this section let $F : \mathbb{C}^n \to \mathbb{C}^m$ be a polynomial system consisting of m polynomials in n unknowns with $m \geq n$. We denote by $J(x) \in \mathbb{C}^{m\times n}$ the Jacobian of F at $x \in \mathbb{C}^n$. We also assume that all considered Jacobian matrices $J(x)$ have full column rank n. The Newton iterations are

$$\begin{aligned} \Delta x^k &= -J(x^k)^\dagger F(x^k) \\ x^{k+1} &= x^k + \Delta x^k \end{aligned} \quad , \quad k = 0, 1, 2, \ldots \tag{2.1}$$

where $J(x^k)^\dagger$ denotes the Moore-Penrose pseudo-inverse of the Jacobian. Note that in the case $m = n$ the pseudo-inverse is identical with the ordinary matrix inverse. In the case $m > n$ the iteration (2.1) is also referred to as the *Gauß-Newton* iteration. We want to remark that in practice the inverse is not computed explicitly, rather the update Δx^k is computed by first factorizing $J(x^k)$ and then solving a simpler linear system.

An important property of Newton's method is that it is invariant under certain affine transformations. That is, for non-singular matrices $A \in \mathbb{C}^{m\times m}$, respectively unitary matrices in the case $m > n$, the transformed system $AF(x)$ results in exactly the same

Newton iterates given the same start value. This property of Newton's method is referred to as *affine covariance* [Deu74, Deu11]. Although this a fundamental property of Newton's method, it only appears in few convergence theorems for local Newton methods. The first systematic approach towards affine covariance has been started by Deuflhard in [Deu74] where it is called "affine invariance". Affine covariant convergence theorems are desirable in the context of path tracking since they lead to results in terms of iterates $\{x^k\}$, correction norms $\|\Delta x^k\|$ or error norms $\|x^k - x^*\|$.

2.1. **An affine covariant convergence theorem.** We present a generalized proof of the convergence theorem of Mysovskikh [Mys49] by Deuflhard and Heindl [DH79]. This result is also referred to as a (refined) Newton-Mysovskikh theorem. The following is a specialized version of the more general Theorem 4 in [DH79].

Theorem 2.1 (Newton-Mysovskikh)**.** *Let F be defined as above. Assume that there exists an open convex subset D_0 of $\mathbb{C}^n$, $x^0 \in D_0$, and constants $\alpha, \omega \geq 0$ such that*

(1) *$J(x)$ has full column rank for all $x \in D_0$,*
(2) *$\|J(x^0)^\dagger F(x^0)\| \leqq \alpha$,*
(3) *$\|J(y)^\dagger (J(x + t(y - x)) - J(x))\| \leq \omega t \|y - x\|$ for all $x, y \in D_0$ and $t \in [0, 1]$,*
(4) *$h := \frac{1}{2}\alpha\omega < 1$,*
(5) *$S := \{z \in \mathbb{C}^n \mid \|z - x^0\| < t^*\}$ with $t^* := \frac{\alpha}{1-h}$.*

Then, the iterates (2.1) are well defined, remain in $\bar{S}$ and converge to some $x^* \in \bar{S}$ with

$$J(x^*)^\dagger F(x^*) = 0 \; . \tag{2.2}$$

Furthermore, convergence can be estimated according to

$$\|\Delta x^k\| \leq \frac{1}{2}\omega \|\Delta x^{k-1}\|^2 \; . \tag{2.3}$$

The condition (2.2) is equivalent to finding a solution x^* with minimal Euclidean norm. In particular, in our context we can assume that $F(x^*) = 0$. Then we can state the following corollary which is a specialized version of Theorem 4.8 in [Deu11].

Corollary 2.2. *Under the assumptions of Theorem 2.1 suppose x^* satisfies $F(x^*) = 0$. Let*

$$\sigma := \|x^0 - x^*\| < \frac{2}{\omega} =: \bar{\sigma} \; .$$

Then the following holds:

(1) For any starting point $x^0 \in S(x^*, \bar{\sigma})$, the Newton iterates remain in $\bar{S}(x^*, \sigma)$ and converge to x^* at the estimated rate

$$\|x^{k+1} - x^*\| \leq \frac{1}{2}\omega \|x^k - x^*\|^2 \; .$$

(2) The solution x^* is unique in the open ball $S(x^*, \bar{\sigma})$.

(3) The following error estimate holds

$$\|x^k - x^*\| \leq \frac{\|\Delta x^k\|}{1 - \frac{1}{2}\omega\|\Delta x^k\|} \,. \tag{2.4}$$

2.2. **A termination criterion.** We now want to use the theoretical results from the convergence analysis in Theorem 2.1 and Corollary 2.2 to construct a termination criterion for Newton's method. This follows the approach by Deuflhard in [Deu11].

We monitor the contraction condition

$$\Theta_{k-1} := \frac{\|\Delta x^k\|}{\|\Delta x^{k-1}\|} \tag{2.5}$$

in terms of the Newton corrections. From Theorem 2.1 we know that

$$\Theta_k \leq \frac{1}{2} h_k \tag{2.6}$$

where $h_k := \omega\|x^{k+1} - x^k\|$. A desirable criterion to terminate the Newton iteration is

$$\|x^k - x^*\| \leq \tau$$

where τ is a user provided error tolerance. With (2.6) we get a computationally available estimate $[h_k] := 2\Theta_k \leq h_k$ and by combining this with the error estimate (2.4), we get the *a-posteriori* termination criterion

$$\frac{\|\Delta x^k\|}{1 - \Theta_k} \leq \tau \,. \tag{2.7}$$

A drawback of this criterion is that for the computation of Θ_k we also need to know x^{k+2}. Thus, we now want to derive an *a-priori* estimate for h_k. By multiplying ω with both sides of (2.3) and using the assumption $\Theta_{k-1} < \frac{1}{2}$ we obtain

$$[h_k] \leq \Theta_{k-1}[h_{k-1}] = 2\Theta_{k-1}^2 \leq h_k \,.$$

We arrive at the *a-priori* termination criterion

$$\frac{\|\Delta x^k\|}{1 - 2\Theta_{k-1}^2} \leq \tau \,. \tag{2.8}$$

As a heuristic against *path jumping* we only allow $N+1$ Newton iteration steps where N is usually 1 or 2. To be precise, usually the Newton iteration is successfully terminated if $\|\Delta x^N\| \leq \tau$ and x^{N+1} is returned. But as we have seen above, this would roughly measure the accuracy of x^N and *not* of x^{N+1} – the last Newton iteration is primarily used for assessing the accuracy of the previous iterate. We therefore propose to use instead a *simplified* Newton iterate to assess the accuracy of x^N, that is we reuse the previous Jacobian to compute the Newton update

$$\overline{\Delta x}^k := J(x^{k-1})^\dagger F(x^k) \,.$$

If x^{k-1} is already close to the solution x^* then $\overline{\Delta x^k}$ and Δx^k will be very similar. Defining a modified contradiction condition

$$\overline{\Theta}_{k-1} := \frac{\|\overline{\Delta x^k}\|}{\|\Delta x^{k-1}\|}$$

and adopting the a-priori termination criterion (2.8) we arrive at the termination criterion

$$\frac{\|\overline{\Delta x}^N\|}{1 - 2\overline{\Theta}_{N-1}^2} \leq \tau \ . \tag{2.9}$$

3. An Adaptive Step Size Control

For an efficient path tracking routine it is important to choose the step sizes as large as possible while minimizing the risk of path jumping. So far, most polynomial homotopy continuation software, e.g., [BHSW], relies on the following adaptive step size control: if a step with step size Δt got rejected, the step is tried again with the reduced step size $\Delta t = a\Delta t$ for some $a \in (0, 1)$. If there have been $M > 0$ successes in a row, the step size is expanded by $\Delta t = \Delta t/a$. In [BHSW] the authors choose the parameters $a = 1/2$ and $M = 5$. As already noted at the end of Section 2, in order to decrease the risk that path jumping occurs Newton's method is usually restricted to only 2 or 3 iterations.

In the following we derive a new adaptive step size control which improves upon the existing paradigm by incorporating more local information of the path as well as the allowed number of Newton iterations and the chosen predictor. This new step size control can be seen as an adaption of a step size control developed by Deuflhard in his habilitation thesis [Deu79] and also presented in the book [Deu11].

We assume that for all $t \in [0, 1]$ the $m \times n$ Jacobian $H_x(x(t), t)$ has full column rank n. An adaptive step size control should take the properties of the predictor into account. To be precise, given some $x(t_k) \in \mathbb{C}^n$, a predictor allows us to construct a *prediction path* $\widehat{x}(t)$ which locally approximates $x(t)$ for $t \geq t_k$. For example, the tangent $\dot{x}(t)$ can be computed by

$$\dot{x}(t) = -H_x(x(t), t)^\dagger \, H_t(x(t), t) \tag{3.1}$$

and for $t \geq t_k$ the prediction path $\widehat{x}(t) = x(t_k) + (t - t_k)\dot{x}(t_k)$ is the *tangent* or *Euler predictor*. There are many more possible predictors and we classify them by their *order*.

Definition 3.1 (Order of a predictor)**.** Let $\Delta t := t - t_k$. A continuation method defined via the prediction path $\widehat{x}(t)$ is said to be of order p if a constant η_p exists such that

$$\|x(t) - \widehat{x}(t)\| \leq \eta_p \Delta t^p \ . \tag{3.2}$$

⬡

Example 3.2. Let $t_k = 0$ and $\Delta t = t$. For the tangent predictor we obtain

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

Thus, the tangent predictor is of order $p = 2$. ⬡

3.1. **Feasible step sizes.** The *local step sizes* $\Delta t_k := t_{k+1} - t_k$ have to be chosen such that the Newton method starting at the predicted value $\widehat{x}(t_{k+1})$ achieves within $N+1$ iterations an accuracy of τ towards the solution $x(t_{k+1})$ on the solution. That is, given a desired accuracy τ the N-th Newton iterate x^N with $x^0 = \widehat{x}(t_{k+1})$ has to satisfy

$$\|x^N - x(t_{k+1})\| \leq \tau \, .$$

For this we start with a theoretical analysis of *feasible step sizes* which we then can use to derive an adaptive step size control. This is the main theoretical result of this article.

Theorem 3.3. *Let $\widehat{x}(t)$ denote a prediction method of order p as defined in (3.2) based on the previous solution point $x(t_k)$. Fix $t^* > t_k$ and let $D \subset \mathbb{C}^m$ be an open convex set containing $\widehat{x}(t)$ and $x(t)$ for all $t \in [t_k, t^*\,]$. Assume that there exist a constant $\omega \geq 0$ such that the Lipschitz-condition*

$$\|H_x(v,t)^\dagger (H_x(u + s(v-u), t) - H_x(u,t))\| \leq \omega s \|v - u\| \tag{3.3}$$

holds for all $t \in [t_k, t^*\,]$, $s \in [0,1]$ and $u, v \in D$. Then the ordinary Newton method with starting point $\widehat{x}(t_{k+1})$ achieves within $N+1$ iterations an accuracy of at least τ towards the solution point $x(t_{k+1})$ for all step sizes

$$\Delta t_k \leq \Delta t_{\max} \tag{3.4}$$

where

$$\Delta t_{\max} := \min\left\{ \left(\frac{\sqrt{4\delta_{N,\omega} + 1} - 1}{\omega \eta_p} \right)^{1/p}, t^* - t_k \right\}$$

and $\delta_{N,\omega} := \min\{ \sqrt{\tfrac{\omega}{2}} (\tau/(1 + \tfrac{\omega}{2}\tau))^{\frac{1}{2N}}, 1 \}$.

Proof. By the error estimate (2.4) in Corollary 2.2 of the Newton-Mysovskikh Theorem 2.1 it is sufficient to achieve

$$\frac{\|\Delta x^N\|}{1 - \frac{\omega}{2}\|\Delta x^N\|} \leq \tau \, ,$$

or equivalently

$$\|\Delta x^N\| \leq \frac{\tau}{1 + \frac{\omega}{2}\tau} \, . \tag{3.5}$$

Using (2.3) we also obtain the estimate

$$\|\Delta x^N\| \leq \frac{1}{2}\omega \|\Delta x^{N-1}\|^2 \leq \left(\frac{\omega}{2}\right)^N \|\Delta x^0\|^{2N} \tag{3.6}$$

under the assumption $\alpha(t)\omega \leq 2$. Therefore, we achieve the desired accuracy in N steps, provided that

$$\sqrt{\frac{\omega}{2}} \|\Delta x^0\| \leq \left(\frac{\tau}{1 + \frac{\omega}{2}\tau} \right)^{\frac{1}{2N}} \tag{3.7}$$

So an upper bound for the first Newton correction $\|\Delta x^0(t)\|$ needs to be derived. To obtain this, we estimate

$$\begin{aligned}\|\Delta x^0(t)\| &= \|H_x(\widehat{x}(t),t)^\dagger H(\widehat{x}(t),t)\| = \|H_x(\widehat{x},t)^\dagger (H(\widehat{x},t) - H(x,t))\| \\ &= \|H_x(\widehat{x}(t),t)^\dagger \int_0^1 H_x(x+s(\widehat{x}-x),t)(\widehat{x}-x)ds\| \\ &\leq \|\widehat{x}-x\|(1+\frac{1}{2}\omega\|\widehat{x}-x\|)\end{aligned}$$

where a detailed proof of the last inequality can be found in [Deu79, Thm. 4].

With the order of a prediction method we are now able to conclude

$$\|\Delta x^0(t)\| \leq \alpha(t) \leq \eta_p t^p (1+\frac{\omega}{2}\eta_p t^p)\,. \tag{3.8}$$

By combining (3.7) and (3.8) with the requirement $\frac{1}{2}\alpha(t)\omega \leq 1$ we arrive at

$$\frac{\omega}{2}\|\Delta x^0\| \leq \frac{\omega}{2}\eta_p t^p(1+\frac{1}{2}\omega\eta_p t^p) \leq \min\left\{\sqrt{\frac{\omega}{2}}\Big(\frac{\tau}{1+\frac{\omega}{2}\tau}\Big)^{\frac{1}{2N}}, 1\right\} = \delta_{N,\omega} \tag{3.9}$$

from which follows

$$\omega\eta_p t^p \leq \sqrt{4\delta_{N,\omega}+1} - 1$$

and thus also the inequality (3.4). □

3.2. **A prediction and correction strategy.** The analysis of feasible step sizes in Theorem 3.3 gives us an indication on how to choose the local step sizes $\Delta t_k := t_{k+1} - t_k$ such that the Newton iteration starting at the predicted value $\widehat{x}(t_{k+1})$ converges to the solution $x(t_{k+1})$ on the solution path with at most $N+1$ iterations. But since we do not know the theoretical quantities η_p and ω we need to replace them with computational available estimates $[\omega] \approx \omega$ and $[\eta_p] \approx \eta_p$. Then, we arrive at the maximal feasible step size estimates

$$[\Delta t_{\max}] := \Big(\frac{\sqrt{4\delta_{N,[\omega]}+1}-1}{[\omega][\eta_p]}\Big)^{1/p} \approx \Delta t_{\max}\,. \tag{3.10}$$

Due to the fact that we only have computational estimates of η_p and ω it is possible that the computed step size is too large. This is $[\Delta t_{\max}] \geq \Delta t_{\max}$. Then, it can happen that the desired accuracy τ is not achieved within $N+1$ iterations and we have to reject the step $t_k \to t_k + \Delta t_k$ and repeat it with a reduced step size $\Delta t'_k$. Thus, we need a *prediction strategy* to compute a new step size Δt_{k+1} after a step got accepted but also a *correction strategy* in the case that a step got rejected.

From the convergence analysis of the generalized Newton method we know that

$$\Theta_0(t) = \frac{\|\Delta x^1(t)\|}{\|\Delta x^0(t)\|} \leq \frac{1}{2}\omega\|\Delta x^0(t)\|\,. \tag{3.11}$$

Combined with (3.9) and the monotone increasing function

$$g(\tau) := \sqrt{4\tau+1} - 1$$

this yields

$$g(\Theta_0(t)) \leq \omega \eta_p t^p \leq g(\delta_{N,\omega}) \; . \tag{3.12}$$

Assume now we have computed $m \leq N+1$ Newton iterations for an actual step size Δt_k. Then we can obtain an estimate for ω by

$$[\omega] := \max_{k=1,\ldots,m-1} 2 \frac{\|\Delta x^k\|}{\|\Delta x^{k-1}\|^2} \leq \omega \tag{3.13}$$

and for η_p by (3.12)

$$[\eta_p] := \frac{g(\Theta_0)}{[\omega] \Delta t_k^p} \; .$$

We arrive at the step size *correction strategy*

$$\Delta t_k' = \left(\frac{g(\delta_{N,[\omega]})}{g(\Theta_0)} \right)^{1/p} \Delta t_k \; . \tag{3.14}$$

If $\Theta_0 > \delta_{N,[\omega]}$, this is clearly a step size restriction. But in some cases it still can happen that the step got rejected although $\Theta_0 < \delta_{N,[\omega]}$. In this case we simply half the step size, that is we set $\Delta t_k' = \frac{1}{2} \Delta t_k$.

In order to derive a *prediction strategy* we need to derive *a-priori* estimates ω and η_p. Since the Newton iteration was successful with a solution $\bar{x}(t_{k+1})$ we can use the definition of the order of a predictor to compute an estimate for η_p by

$$[\eta_p]_{k+1} := \frac{\|\widehat{x}(t_{k+1}) - \bar{x}(t_{k+1})\|}{\Delta t_k^p} \; . \tag{3.15}$$

We then arrive at the prediction strategy

$$\Delta t_{k+1} = \mu \left(\frac{g(\delta_{N,[\omega]})}{[\omega] \|\widehat{x}(t_{k+1}) - \bar{x}(t_{k+1})\|} \right)^{1/p} \Delta t_k \; . \tag{3.16}$$

where $\mu \in (0,1]$ is an additional scaling factor. Considering that the new step size is modeled after the maximal feasible step size it proves beneficial to reduce it by the fixed factor μ. In our experience a factor of $\mu = 0.9$ works well.

Since N is usually quite small, that is $N = 1$ or $N = 2$, it is important for the prediction strategy to not underestimate ω and η_p by too much. We therefore want to improve the estimates by introducing prediction methods. By modeling η_p as a linear function

$$\widehat{\eta}_p(t) = [\eta_p]_{k+1} + t \frac{[\eta_p]_{k+1} - [\eta_p]_k}{\Delta t_k}$$

and assuming $\Delta t_{k+1} = \Delta t_k$, we arrive at

$$[\widehat{\eta}_p]_{k+1} := \max\{2[\eta_p]_{k+1} - [\eta_p]_k, [\eta_p]_{k+1}\} \; .$$

Similarly, a simple linear predictor for ω can be derived.

4. Path Tracking in Projective Space

The numerical path tracking problem (1.1) can be cast into the context of complex projective space $\mathbb{P}^n = \mathbb{P}(\mathbb{C}^{n+1})$ rather than in Euclidean space $\mathbb{C}^n$ by *homogenizing* the homotopy H with respect to the variables $x_1, \ldots, x_n$. The projective space $\mathbb{P}^n$ is the set of all nonzero vectors $x, y \in \mathbb{C}^{n+1}$ modulo the equivalence relation $x = \lambda y$ for $\lambda \neq 0$. This has multiple advantages all related to the fact that $\mathbb{P}^n$ is a compactification of $\mathbb{C}^n$. In particular, for a well constructed homotopy in $\mathbb{P}^n$, all solution paths only have finite length [Mor86].

4.1. **Affine coordinate patches.** There is a choice of selecting a representative for each point in $\mathbb{P}^n$. We fix representatives by choosing in each step t an affine coordinate patch, that is we pick a point $v_t \in \mathbb{C}^{n+1}$ and impose the additional constraint

$$\langle x(t), v_t \rangle = 1$$

where $\langle \cdot, \cdot \rangle$ is the usual Euclidean inner product in $\mathbb{C}^{n+1}$. In [Mor86] it is proposed to perform all computations on one fixed random affine coordinate patch by picking a random $v \in \mathbb{C}^{n+1}$ and setting $v_t = v$ for all t. Then, the tangent vectors $\dot{x}(t)$ satisfy $\langle \dot{x}(t), v \rangle = 0$, i.e., Newton updates are performed in the linear space $T_v := \{z \in \mathbb{C}^{n+1} \mid \langle v, z \rangle = 0\}$. The fixed random affine coordinate patch preserves with probability one the property that all solution paths have finite length. A disadvantage of this approach is that ill-conditioning could be introduced artificially. Nonetheless, due to its simplicity this is used in many implementations.

Another idea is to adaptively change the affine coordinate patch during the path tracking. An approach with a particular nice geometric interpretation is referred to as the *orthogonal patch*. It is defined by setting $v_t = x(t)$. Then,

$$\langle x(t), x(t) \rangle = 1 \quad \text{and} \quad \langle \dot{x}(t), x(t) \rangle = 0 \, .$$

Here, Newton updates are performed in $T_{x(t)}$. In fact, $T_{x(t)}$ is the appropriate model for the tangent space $T_{x(t)}\mathbb{P}^n$, see [SS93]. If we change the affine coordinate patch not only at each t_k but after each Newton iteration, we arrive at a projective version of Newton's method [BC13]. In general the orthogonal patch results in better conditioned paths than the fixed random affine patch approach [CL12, HR18].

4.2. **Adaptive step size control in projective space.** The adaptive step size control from Section 3 is developed for affine space and not projective space. If we work with a fixed random affine patch, we effectively work on one affine chart of the projective space and the step size control is directly applicable. If we work with dynamically changing affine patches, we have to be a little bit more careful. In particular we have to make sure that the computational estimates for ω and η_p obtained on one affine chart are still good estimates with respect to the new chart. Also there is a choice on how often the affine chart should be changed. When working with the orthogonal patch, a possible choice is to change the patch before each Newton step which would resemble the projective Newton's method. But this conflicts with the affine Newton-Mysovskikh Theorem 2.1 and it is not clear that the step size control is still valid. We therefore propose instead to update the patch only after each accepted step. This still guarantees at each step the applicability

of Theorem 2.1 and in our experiments the accuracy of the computational estimates for ω and η_p is still satisfying.

5. Implementation details and computation experiments

HomotopyContinuation.jl [BT18] is a software package for solving polynomial system by homotopy continuation methods currently under active development by Paul Breiding and the author. HomotopyContinuation.jl is written in the programming language Julia [BEKS17]. All the examples discussed here were run on an Intel Core i5-7500 3.4 GHz processor running MacOS 10.14.2 with Julia 1.0.3.

5.1. **Predictors.** The step size control from Section 3 gives freedom in the choice of predictor. In [BHS11] it is reported that for the adaptive precision path tracking algorithm [BHSW08] (embedded) Runge-Kutta methods of higher order significantly outperform the tangent predictor. Similarly, HomotopyContinuation.jl used up to version 0.4 the classic 4^{th}-order Runge-Kutta method (RK4 for short). Note that what is referred to as Runge-Kutta method of order m in the ordinary differential equation literature would be a predictor of order $m+1$ by Definition 3.2. One downside of Runge-Kutta methods is that they are fairly expensive. The Runge-Kutta methods up to order $m \leq 4$ need m evaluations of the right hand side of (3.1), each involving a computation of the derivatives and the solution of a linear system. There is also the tradeoff of less steps per path against the computational cost of each step to be considered.

We also experimented with predictors based on a Taylor or Padé approximation by computing higher derivatives $\ddot{x}(t_k), \dddot{x}(t_k), \ldots$ because they can be approximated by few additional evaluations of the homotopy using the techniques developed in [Mac89]. While Taylor an Padé approximations perform better than the simple tangent predictor they are still outperformed by Runge-Kutta methods in our experiments.

At a time step t_k almost every predictor method needs the derivative $\dot{x}(t_k)$. An easy optimization to save some computational resources is as follows: If a step with step size Δt_k gets rejected, there is no need to recompute the derivate $\dot{x}(t_k)$ for the next step. Thus, after a step rejection a prediction costs one evaluation less of the right hand side. After implementing this caching strategy Heun's method, a Runge-Kutta method of order 2, slightly outperforms RK4.

In order to compare the performance of different predictors we pick a range of real-world polynomial systems of different types, presented in Table 1 [Ver]. The results are depicted in Figure 5.1.

5.2. **Comparison of adaptive step size controls.** To demonstrate the effectiveness of the new adaptive step size control we compare in Table 2 the number of accepted and rejected steps of the simple step size control introduced at the beginning of Section 3 and the new adaptive step size control developed in this paper. We see that the number of accepted steps decrease significantly with only a slight increase in the number of rejected steps.

In this paper we also introduce two more minor performance improvements for the path tracking algorithm. A cheaper a-priori termination criterion (2.9) for Newton's method

TABLE 1. Overview of the polynomial systems chosen for the comparison. In the characteristics n is the number of unknowns, D is the Bézout number of the system and MV is the mixed volume. The system were taken from the database by Jan Verschelde [Ver].

Polynomial systems			Characteristics			# Roots	
Name	Description	Ref	n	D	MV	$\mathbb{C}$	$\mathbb{R}$
cyclic7	The cyclic 7-roots problems	[BF91]	7	5,024	924	924	56
ipp2	The 6R inverse position problem	[MW91]	11	1,024	288	16	0
katsura11	A problem of magnetism in physics	[Kat90]	12	2,048	2,048	2,048	326

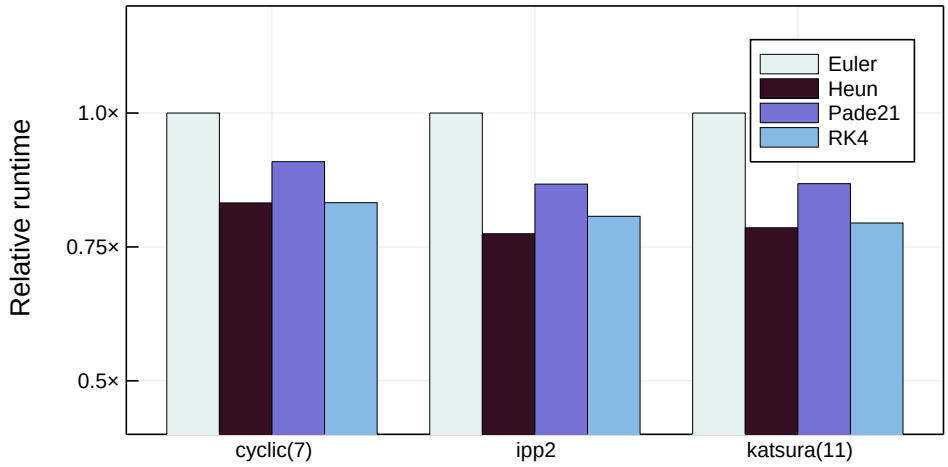

FIGURE 1. Comparison of the average runtime for solving three different system using different predictors where the average is obtained by making 100 runs. The runtime is normalized such that the Euler predictor is 1.0. Pade21 refers to a (2,1) Padé approximation.

and the idea to cache the tangent vectors $\dot{x}(t_k)$. These improvements are implemented in HomotopyContinuation.jl version 0.5. We compare the runtime for the polynomial systems in Table 1 between HomotopyContinuation.jl v0.5 and v0.4.3, the latest release without these improvements. For reference we also compare these runtimes with Bertini [BHSW] using only double precision arithmetic. The results are presented in Table 3 and are in accordance with what one expects from the results in Table 2, a slightly higher performance increase due to the other improvements. Also see [BT18] for another comparison of homotopy continuation packages for the systems in Table 1.

6. CONCLUSION

In this paper we introduced a new adaptive step size control for path tracking in the context of homotopy continuation methods. This step size control incorporates information about the local geometry of the problem as well as the used predictor method. The

TABLE 2. Comparison of the average number of steps per path necessary to track a total degree straight line homotopy from $t = 1.0$ to the start of the endgame zone at $t = 0.1$. The 'old' columns refer to the simple step size control introduced at the beginning of Section 3 and 'new' is the adaptive step size control developed in this paper. The paths were tracked with a tolerance of $\tau = 10^{-7}$, with at most 3 Newton iterations and Heun's method as a predictor. The average is obtained by making 100 runs.

	Accepted steps		Rejected steps		Total steps		
System	old	new	old	new	old	new	new / old
cyclic7	30.44	21.14	5.42	6.89	35.86	28.03	0,78
ipp2	33.84	22.80	5.95	7.69	39.78	30.45	0.77
katsura11	38.69	22.88	6.12	8.33	44.81	31.20	0.69

TABLE 3. Average of the total runtime (including endgame) for solving the polynomial systems in Table 1. The average is obtained by making 100 runs. v0.4 and v0.5 refer to HomotopyContinuation.jl version 0.4 and version 0.5.

	Runtime in seconds			Relative performance	
System	v0.4	v0.5	Bertini	v0.5 / v0.4	v0.5 / Bertini
cyclic7	2.50	2.03	27.91	0.81	0.07
ipp2	0.67	0.53	5.25	0.79	0.10
katsura11	2.22	1.48	14.08	0.67	0.11

improvement over the previously most commonly used step size control is demonstrated on a range of polynomial systems.

7. Acknowledgements

The author would like to thank Folkmar Bornemann, Paul Breiding, Peter Deuflhard and Michael Joswig for helpful discussions.

References

[BC13] Peter Bürgisser and Felipe Cucker. *Condition: The geometry of numerical algorithms*, volume 349. Springer Science & Business Media, 2013.

[BEKS17] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

[BF91] Göran Björck and Ralf Fröberg. A faster way to count the solutions of inhomogeneous systems of algebraic equations, with applications to cyclic n-roots. *Journal of Symbolic Computation*, 12(3):329–336, 1991.

[BHS11] Daniel J. Bates, Jonathan D. Hauenstein, and Andrew J. Sommese. Efficient path tracking methods. *Numerical Algorithms*, 58(4):451–459, Dec 2011.

[BHSW] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. Bertini: Software for numerical algebraic geometry. Available at bertini.nd.edu with permanent doi: dx.doi.org/10.7274/R0H41PB5.

[BHSW08] Daniel J Bates, Jonathan D Hauenstein, Andrew J Sommese, and Charles W Wampler. Adaptive multiprecision path tracking. *SIAM Journal on Numerical Analysis*, 46(2):722–746, 2008.

[BL13] Carlos Beltrán and Anton Leykin. Robust certified numerical homotopy tracking. *Foundations of Computational Mathematics*, 13(2):253–295, 2013.

[BT18] Paul Breiding and Sascha Timme. HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia. In *International Congress on Mathematical Software*, pages 458–465. Springer, 2018.

[CL12] Tianran Chen and Tien-Yien Li. Spherical projective path tracking for homotopy continuation methods. *Communications in Information and Systems*, 12(3):195–220, 2012.

[Deu74] Peter Deuflhard. A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting. *Numerische Mathematik*, 22(4):289–315, 1974.

[Deu79] Peter Deuflhard. A stepsize control for continuation methods and its special application to multiple shooting techniques. *Numerische Mathematik*, 33(2):115–146, 1979.

[Deu11] Peter Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, volume 35. Springer Science & Business Media, 2011.

[DH79] Peter Deuflhard and Gerhard Heindl. Affine invariant convergence theorems for Newtons method and extensions to related methods. *SIAM Journal on Numerical Analysis*, 16(1):1–10, 1979.

[HR18] Jonathan D Hauenstein and Margaret H Regan. Adaptive strategies for solving parameterized systems using homotopy continuation. *Applied Mathematics and Computation*, 332:19–34, 2018.

[HS97] Richard I Hartley and Peter Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[Kat90] Shlgetoshl Katsura. Spin glass problem by the method of integral equation of the effective field. *New Trends in Magnetism*, pages 110–121, 1990.

[Mac89] Wolfgang Mackens. Numerical differentiation of implicitly defined space curves. *Computing*, 41(3):237–260, 1989.

[Mor86] Alexander P Morgan. A transformation to avoid solutions at infinity for polynomial systems. *Applied mathematics and computation*, 18(1):77–86, 1986.

[Mor87] Alexander Morgan. *Solving Polynominal Systems Using Continuation for Engineering and Scientific Problems*. Prentice-Hall, 1987.

[MW91] Alexander Morgan and Charles Wampler. Solving the 6R inverse position problem using a generic-case solution methodology. *Mechanism and Machine Theory*, 26(1):91–106, 1991.

[Mys49] Ivan Mysovskikh. On the convergence of Newton's method. *Collected works on approximation analysis of the Leningrad Branch of the Institute*, 28:145–147, 1949.

[NTI16] Jatin Narula, Abhinav Tiwari, and Oleg A. Igoshin. Role of autoregulation and relative synthesis of operon partners in alternative sigma factor networks. *PLOS Computational Biology*, 12(12):1–25, 12 2016.

[SS93] Michael Shub and Steve Smale. Complexity of Bezout's theorem I: Geometric aspects. *Journal of the American Mathematical Society*, 6(2):459–501, 1993.

[SSN05] Henrik Stewenius, Frederik Schaffalitzky, and David Nister. How hard is 3-view triangulation really? In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 686–693. IEEE, 2005.

[SW05] Andrew J Sommese and Charles W Wampler. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. WORLD SCIENTIFIC, 2005.

[Ver] Jan Verschelde. The database of polynomial systems. Data retrieved from the database of Jan Verschelde available at, http://homepages.math.uic.edu/~jan/. Accessed: 2019-01-15.

[WS11] Charles W. Wampler and Andrew J. Sommese. Numerical algebraic geometry and algebraic kinematics. *Acta Numerica*, 20:469567, 2011.

Sascha Timme, Technische Universität Berlin, Institut für Mathematik, Strasse des 17. Juni 136, 10623 Berlin, Germany

E-mail address: timme@math.tu-berlin.de