Int. J. Appl. Comput. Math (2020) 6:120
https://doi.org/10.1007/s40819-020-00868-5

ORIGINAL PAPER

Adaptive Newton-Type Schemes Based on Projections

Mario Amrein[1] · **Norbert Hilber**[1]

Published online: 29 July 2020

Abstract
In this work we present and discuss a possible globalization concept for Newton-type methods. We consider nonlinear problems $f(x) = 0$ in $\mathbb{R}^n$ using the concepts from ordinary differential equations as a basis for the proposed numerical solution procedure. Thus, the starting point of our approach is within the framework of solving ordinary differential equations numerically. Accordingly, we are able to reformulate general Newton-type iteration schemes using an adaptive step size control procedure. In doing so, we derive and discuss a discrete adaptive solution scheme, thereby trying to mimic the underlying continuous problem numerically without losing the famous quadratic convergence regime of the classical Newton method in a vicinity of a regular solution. The derivation of the proposed adaptive iteration scheme relies on a simple orthogonal projection argument taking into account that, sufficiently close to regular solutions, the vector field corresponding to the Newton scheme is approximately linear. We test and exemplify our adaptive root-finding scheme using a few low-dimensional examples. Based on the presented examples, we finally show some performance data.

Keywords Newton-type methods · Vector fields · Adaptive root finding · Nonlinear equations · Globalization concepts · Continuous Newton method

Mathematics Subject Classification 37N30 · 46N40 · 65H10 · 37B25

Introduction

In this note, we are interested in the problem: Find $x_\infty \in \mathbb{R}^n$ such that

$$f(x_\infty) = 0,$$

where $f : \Omega \to \mathbb{R}^n$ denotes a possibly nonlinear function defined on the open subset $\Omega \subset \mathbb{R}^n$. Of course this problem is one of the well known and possibly most addressed issues in numerical mathematics and has been studied by several authors in the past. Here we study the problem of computing the roots of f numerically. For $x \in \Omega$ let the map $x \mapsto \mathsf{A}(x) \in \mathbb{R}^{n\times n}$ be continuous. Next we set $\mathsf{F}(x) := \mathsf{A}(x) f(x)$ and concentrate on the initial value problem

✉ Mario Amrein
mario.amrein@zhaw.ch

[1] ZHAW, Technoparkstrasse 2, 8400 Winterthur, Switzerland

$$\begin{cases} \dot{x}(t) = \mathsf{F}(x(t)), & t \geq 0, \\ x(0) = x_0, & x_0 \in \Omega. \end{cases} \tag{1}$$

Assuming that a solution $x(t)$ of (1) exists for all $t \geq 0$ with $x(t) \in \Omega$, i.e. $x_\infty := \lim_{t\to\infty} x(t) \in \Omega$ and provided that $\mathsf{A}(x_\infty) f(x_\infty) = 0$ implies $f(x_\infty) = 0$, we can try to follow the solution $x(t)$ numerically to end up with an approximate root for f. In actual computations however, apart from trivial problems, we can solve (1) only numerically. The simplest routine is given by the *forward* Euler method. More precisely: For an initial value $x_0 \in \Omega$, a simple discrete version of the initial value problem (1) is given by

$$x_{n+1} = x_n + t_n \mathsf{F}(x_n), \quad t_n \in (0, 1], \quad n \geq 0. \tag{2}$$

Obviously, depending on the non-linearity of F and the choice of the initial value x_0, such an iterative scheme is more or less meaningful for $n \to \infty$. Indeed, supposing that the limit for $n \to \infty$ of the sequence $(x_n)_{n\geq 0}$ generated by (2) exists, we end up with $\mathsf{F}(x_n) \approx 0$ for n being sufficiently large. Of course, we want to choose F in such a way that the iteration scheme in (2) is able to transport an initial value arbitrarily close to a root of F.

For the remainder of this work, we assume that for all x_n generated by the iteration procedure from (2), there exists a neighborhood of x_n such that the matrix $\mathsf{A}(x)$ is invertible. Let us briefly address some different choices for F. A possible iteration scheme is based on $\mathsf{A}(x) := -\mathsf{Id}$ leading to a fixed point iteration which is also termed *Picard iteration*. It is well known that under certain—quite strong—assumptions on f this scheme converges exponentially fast; see, e.g., [17]. Another interesting choice for F is given by $\mathsf{A}(x) := -\mathsf{J}_f(x)^{-1}$, leading to

$$\mathsf{F}(x) := -\mathsf{J}_f(x)^{-1} f(x). \tag{3}$$

The choice (3) for F in the iteration scheme (2) implies another well established iteration procedure called *Newton's method* with damping. Here for $x \in \Omega$ we denote by $\mathsf{J}_f(x)$ the Jacobian of f at x. Evidently, this method requires reasonably strong assumptions with respect to the differentiability of f as well as the invertibility of the Jacobian $\mathsf{J}_f(x_n)$ for all possible iterates x_n occurring during the iteration procedure. On the other hand and on a *local* level, Newton's method with step size $t_n \equiv 1$ is often celebrated for its quadratic convergence regime 'sufficiently' close to a regular root of f. Also well known are so called *Newton-like methods*, where the Jacobian $\mathsf{J}_f(x)$ is replaced by a continuous approximation. A possible realization of such a method is given by setting $\mathsf{A}(x) := -\mathsf{J}_f(x_0)^{-1}$, i.e., the initial derivative of f will be fixed throughout the whole iteration procedure. The iteration scheme (2) based on various choices for $\mathsf{F}(x)$, where $\mathsf{A}(x)$ typically represents a (continuous) approximate of $-\mathsf{J}_f(x)^{-1}$, has been studied extensively by many authors in the recent past; see, e.g., [4,5,11,12]. Moreover, it is noteworthy that solving (1) with $\mathsf{F}(x) := -\mathsf{J}_f(x)^{-1} f(x)$ on the right is also known as the *continuous Newton method*. A pure analysis studying the long-term behavior of solutions for (1) which possibly lead to a solution of F has also been studied in [8–10,14,15]. Let us remark further that there is a wide research area where various methods are applied which are based on continuous Newton-type methods from (1) and its discrete analogue (2). The goal of the present work is not to give a complete summary of the wide-ranging theory and existing approaches for solving (1) within the context of a root finding procedure, but rather to illustrate some specific properties of vector fields F in order to understand the efficiency of the classical *continuous Newton method* and thereby derive a simple and efficient adaptive numerical solution procedure for the numerical solution of the equation $f(x) = 0$. In summary, in this work we use the fact that close to regular solutions x_∞ the map $x \mapsto -\mathsf{J}_f(x)^{-1} f(x)$ is locally approximately affine linear. Based on this insight,

the novel contribution of this note is the use of the orthogonal projection of a single iteration step resulting from the *forward* Euler scheme onto the discretized global flow. As we will see, this approach is able to retain the quadratic convergence regime of the standard *Newton method* close to the root x_∞ and at the same time the chaotic behavior of the standard *Newton method* will be reduced to a certain extent. Although we only discuss the finite dimensional case, it is noteworthy that the following analysis extends without difficulty to the infinite dimensional Hilbert space case.

Notation

In the main part of this paper, we suppose that—at least—there exists a zero $x_\infty \in \Omega$ solving $f(x_\infty) = 0$, where Ω denotes some open subset of the Euclidean space $\mathbb{R}^n$. In addition, for any two elements $x, y \in \mathbb{R}^n$ we signify by $\langle x, y\rangle$ the standard inner product of $\mathbb{R}^n$ with the corresponding Euclidean norm $\langle x, x\rangle := \|x\|^2$. Moreover, for a given matrix $\mathsf{A} \in \mathbb{R}^{n\times n}$ we denote by $\|\mathsf{A}\|$ the sup-norm induced by $\|\cdot\|$ and $\mathsf{Id} \in \mathbb{R}^{n\times n}$ represents the identity matrix. We further denote by $B_R(x) \subset \mathbb{R}^n$ the open ball with center at x and radius $R > 0$. Finally, whenever the vector field f is differentiable, the derivative at a point $x \in \Omega$ is written as $\mathsf{J}_f(x)$, thereby referring to the Jacobian of f at x.

Outline

This paper is organized as follows: In "Vector Fields v.s. Roots of a Function f" section we first discuss the connection between the local and the global aspects of general Newton-type methods. More precisely, we interpret f as a vector field and focus on the local point of view, i.e, the case when an initial value x_0 of the system (1) is 'close' to a zero x_∞ of the vector field f. Secondly, we consider the situation where initial guesses are no longer assumed to be 'sufficiently close' to a zero x_∞ of f. Based on the discussion within the local point of view, we transform the function f such that—at least on a local level—it is reasonable to expect convergence of our iteration scheme. In addition, we revisit the discretization of the initial value problem (1) in "Adaptivity Based on an Orthogonal Projection Argument" section and define—based on the preceding results—an adaptive iteration scheme for the numerical solution of (1). In "Numerical Experiments" section, we present an algorithmic realization of the previously presented adaptive strategy. Finally, we present a series of low dimensional numerical experiments illustrating the performance of the adaptive strategy proposed in this work. Eventually, we summarize our findings in "Conclusions" section.

Vector Fields v.s. Roots of a Function *f*

Local Perspective

In this section we start from a completely naive point of view by asking the following question: Is there a simple choice for the right hand side of (1) that can be used to transport an initial value $x_0 \in \Omega$ arbitrarily close to a root $x_\infty \in \Omega$ of f? The answer to such a question typically depends on how close the initial guess x_0 is chosen with respect to the zero x_∞. Indeed, if we assume that x_0 is 'sufficiently close' to x_∞, it would be preferable that such an initial guess x_0 can be transported straightforwardly and arbitrarily close to the zero x_∞. However, let us remark on the following:

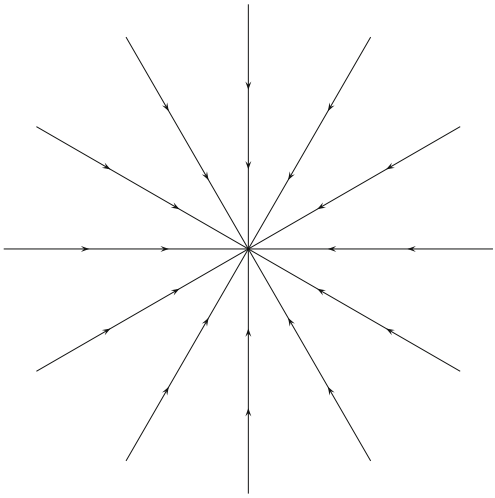

Fig. 1 The direction field associated with $x \mapsto x_\infty - x$. Here, the center of the star signifies x_∞

First of all and for the purpose of simplicity, we suppose that for x_0 'sufficiently' close to x_∞ the function f is affine linear. More precisely, assume that the function f is given by $f(x) := x - x_\infty$ (see also Fig. 1). Thus, if we set $\mathsf{A}(x) := -\mathsf{Id}$ on the right hand side of Eq. (1), the solution is given by $x(t) := x_\infty + (x_0 - x_\infty)\mathrm{e}^{-t}$. Obviously, any initial guess x_0 will be transported arbitrarily close to the zero x_∞. What can we learn from this favorable behavior of $x(t)$? On the one hand it would be preferable that an arbitrary vector field behaves like $\mathsf{F}(x) = x_\infty - x$. In the nonlinear case and from a global perspective, i.e., whenever the initial guess x_0 is far away from a zero x_∞, we would still like to establish a procedure which is able to transport the initial guess into a neighborhood of x_∞, where it is reasonable to assume the previous favorable behavior of the curve $x(t) = x_\infty + (x_0 - x_\infty)\mathrm{e}^{-t}$. So far, our discussion implies that on a local level we can typically expect to find a zero x_∞ whenever $\mathsf{F}(x)$ is close to $x_\infty - x$. Let us therefore transform f in such a way that, at least on a local level, $\mathsf{F}(x) \approx x_\infty - x$ holds (see again Fig. 1).

Global Perspective

As previously discussed, starting in (1) with an initial value $x_0 \in \Omega$, it would be preferable that the root x_∞ is attractive. More precisely, for the initial value x_0 the corresponding solution $x(t)$ should end at x_∞, i.e., $\lim_{t\to\infty} x(t) = x_\infty$ holds. Consequently, we would like to transform the vector field f in such a way that the new vector field—denoted by F—only has zeros which are at least 'locally' attractive. In other words, we want to transform f by $\mathsf{F}(x) := \mathsf{A}(x)f(x)$ such that

$$\mathsf{F}(x) \approx x_\infty - x, \tag{4}$$

holds true, especially whenever x is 'close' to x_∞. A possible choice for F that mimics the map $x \mapsto x_\infty - x$ whenever x is close to the root x_∞ is given by

$$\mathsf{A}(x) := -\mathsf{J}_f(x)^{-1}. \tag{5}$$

Obviously, the price we have to pay for this choice is that f has to be differentiable with invertible Jacobian. Indeed, if f is twice differentiable with bounded second derivative, we observe that

$$\begin{aligned} \mathsf{F}(x) &= \mathsf{F}(x_\infty) + \mathsf{DF}(x_\infty)(x - x_\infty) + \mathcal{R}(x_\infty, x - x_\infty) \\ &= x_\infty - x + \mathcal{R}(x_\infty, x - x_\infty), \end{aligned} \tag{6}$$

with $\|\mathcal{R}(x_\infty, x - x_\infty)\| = \mathcal{O}(\|x - x_\infty\|^2)$. Incidentally, it is well known that as long as the real parts of the eigenvalues of

$$\mathsf{DF}(x_\infty) = \mathsf{D}[\mathsf{A}(x) f(x)]|_{x=x_\infty} = \mathsf{A}(x_\infty)\mathsf{J}_f(x_\infty),$$

are negative, the zero x_∞ is locally attractive; see e.g., [7]. As a result, if $\mathsf{A}(x_\infty)$ is 'sufficiently' close to the inverse of the Jacobian $-\mathsf{J}_f(x_\infty)$, the zero x_∞ might still be locally attractive. For example we can choose $\mathsf{F}(x) := -\mathsf{J}_f(x_0)^{-1} f(x)$ in (2). Generally speaking, whenever $\mathsf{A}(x)$ is 'sufficiently' close to the inverse of $-\mathsf{J}_f(x)$, we still can hope that—especially on a local level—the iteration procedure (2) is well defined and possibly convergent, i.e., $x_n \to x_\infty$ for $n \to \infty$.

Notice that whenever we can fix $\mathsf{A}(x) := -\mathsf{J}_f(x)^{-1}$, the initial value problem given in (1) reads as follows:

$$\begin{cases} \dot{x}(t) = -\mathsf{J}_f(x(t))^{-1} f(x(t)), & t \geq 0, \\ x(0) = x_0, & x_0 \in \mathbb{R}^n. \end{cases} \tag{7}$$

This initial value problem is also termed *continuous Newton's method* and has been studied by several authors in the past; see, e.g., [1,4–6,8,10,13–16]. Let us briefly show an important feature of the *continuous Newton's method*. Suppose that $x(t)$ solves (7). Then it holds that

$$\frac{\mathsf{d}}{\mathsf{d}t} f(x(t)) = -f(x(t)),$$

from where we deduce

$$f(x(t)) = f(x_0)\mathrm{e}^{-t}.$$

Adaptivity Based on an Orthogonal Projection Argument

In this section, we define an iteration scheme for the numerical solution of (1). Based on the previous observations we further derive a computationally feasible adaptive step size control procedure. To this end, we assume that $\mathsf{F}(x) = \mathsf{A}(x) f(x)$ is sufficiently smooth and that x_∞ is a regular root of f, i.e., $f(x_\infty) = 0$ and the inverse of $\mathsf{J}_f(x_\infty)$ exists. Our analysis starts with a second order Taylor expansion of $\mathsf{F}(x)$ around x_∞ given by

$$\mathsf{F}(x) = \mathsf{DF}(x_\infty)(x - x_\infty) + \mathcal{R}_{x_\infty}(x - x_\infty), \quad \left\|\mathcal{R}_{x_\infty}(x - x_\infty)\right\| = \mathcal{O}(\|x - x_\infty\|^2). \tag{8}$$

Next we recall that whenever we are able to choose $\mathsf{A}(x) := -\mathsf{J}_f(x)^{-1}$ there holds

$$\mathsf{F}(x) = \mathcal{L}(x) + \mathcal{R}_{x_\infty}(x - x_\infty),$$

where $\mathcal{L}(x) := x_\infty - x$. In addition, for $x, y \in \mathbb{R}^n$ we consider the orthogonal projection of x onto y given by

$$\mathrm{proj}_y(x) := \frac{\langle x, y\rangle}{\|y\|^2} y. \tag{9}$$

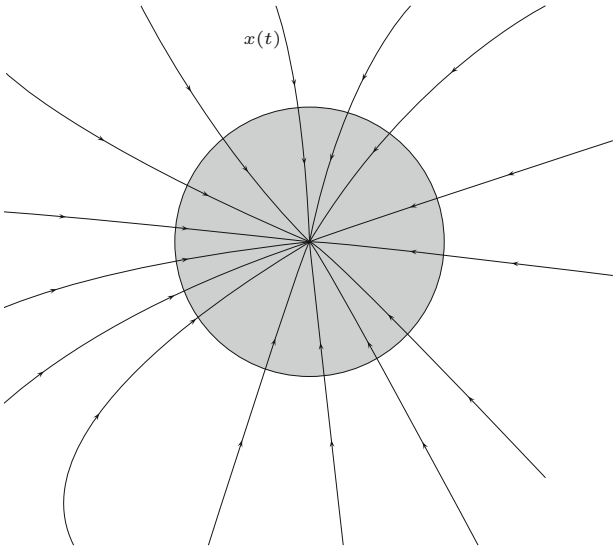

Fig. 2 A neighborhood of x_∞ where the map F behaves like the affine linear map $x \mapsto x_\infty - x$. Note that close to x_∞ the solutions are close to integral curves of the form $x(t) := x_\infty + (x_0 - x_\infty)\mathrm{e}^{-t}$ solving (1)

We now use the orthogonal projection of $\mathsf{F}(x)$ onto $\mathcal{L}(x)$. In particular for $x \neq x_\infty$ Eq. (8) delivers

$$\mathrm{proj}_{\mathcal{L}(x)}(\mathsf{F}(x)) = \frac{\langle \mathsf{F}(x), \mathcal{L}(x)\rangle}{\|\mathcal{L}(x)\|^2} \cdot \mathcal{L}(x) = \mathcal{L}(x) + \frac{\langle \mathcal{R}_{x_\infty}(x - x_\infty), \mathcal{L}(x)\rangle}{\|\mathcal{L}(x)\|^2} \cdot \mathcal{L}(x).$$

Note that in a neighborhood of a regular root x_∞ it holds that $\mathsf{F}(x) \approx \mathcal{L}(x)$ (see Fig. 2). Incidentally, in case of $\mathsf{F}(x) = \mathcal{L}(x)$ there holds $\mathrm{proj}_{\mathcal{L}(x)} = \mathsf{Id}$, the key idea of our proposed approach. For the remainder of this section, we assume that for an initial guess $x_0 \in \Omega$ there exists a solution $x(t)$ for the initial value problem from (1) such that $\lim_{t\to\infty} x(t) = x_\infty$ solves $f(x_\infty) = 0$. Moreover, we assume that there exists an open neighborhood $B_R(x_0) \subset \Omega$ of x_0 such that for all $x \in B_R(x_0)$ there exists a solution $x(t)$ of (1) starting in $x \in B_R(x_0)$ with $\lim_{t\to\infty} x(t) = x_\infty$. Thus, for $t > 0$ being sufficiently small, we can assume that

$$x_1 := x_0 + t\mathsf{F}(x_0) \quad \text{and} \quad x_2 := x_1 + t\mathsf{F}(x_1). \tag{10}$$

are elements of $B_R(x_0)$. We now use $\mathsf{F}(x_0)$ and $\mathsf{F}(x_1)$ and set $v := \mathsf{F}(x_0) + \mathsf{F}(x_1)$. Note that for $t = 1$ and x_2 'close' to x_∞ there holds $v = x_2 - x_0 \approx x_\infty - x_0 = \mathcal{L}(x_0)$. Next we define our effectively computed iterate

$$\tilde{x}_1 := x_0 + t\,\mathrm{proj}_v(\mathsf{F}(x_0)). \tag{11}$$

The situation is depicted in Fig. 3. Note that $\mathsf{F}(x_1) \approx \mathsf{F}(x_0)$ implies

$$\mathrm{proj}_v(\mathsf{F}(x_0)) = \frac{\langle v, \mathsf{F}(x_0)\rangle}{\|v\|^2} v \approx \mathsf{F}(x_0),$$

i.e., $\mathrm{proj}_v(\mathsf{F}(x_0)) \approx \mathsf{Id}$ in this case.

Let us focus on the distance between the exact solution $x(t)$ and its approximate $\tilde{x}_1$ at $t > 0$. In doing so we revisit the proposed approach from [2, §2.3].

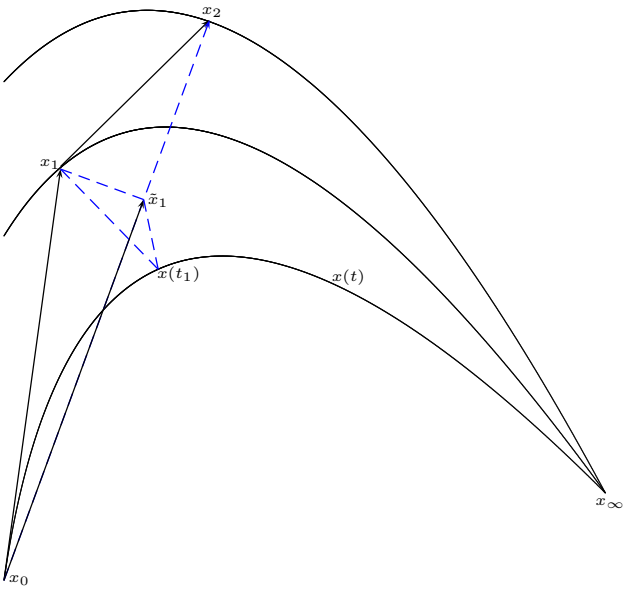

Fig. 3 The projection of $x_1 - x_0 = t_1 \mathsf{F}(x_0)$ onto $t_1 v = x_2 - x_0$ after a time step $t = t_1$

Error Analysis

First we consider the Taylor expansion of $x(t)$ around $t_0 = 0$:

$$\begin{aligned} x(t) &= x_0 + t\dot{x}(0) + t^2 \frac{\ddot{x}(0)}{2} + \mathcal{R}_x(t) \\ &= x_0 + t\mathsf{F}(x_0) + t^2 \frac{\ddot{x}(0)}{2} + \mathcal{R}_x(t) \\ &= x_1 + t^2 \frac{\ddot{x}(0)}{2} + \mathcal{R}_x(t), \quad \text{with} \quad \|\mathcal{R}_x(t)\| = \mathcal{O}(t^3). \end{aligned} \tag{12}$$

Moreover, we will take a look at the expansion of $\mathsf{F}(x)$ around x_1 given by

$$\mathsf{F}(x_1) = \mathsf{F}(x_0) + t\mathsf{DF}(x_0)\mathsf{F}(x_0) + \mathcal{R}_\mathsf{F}(t\mathsf{F}(x_0)) \quad \text{with} \quad \|\mathcal{R}_\mathsf{F}(t\mathsf{F}(x_0))\| = \mathcal{O}(t^2 \|\mathsf{F}(x_0)\|^2).$$

We see that there holds

$$\lim_{t \searrow 0} \frac{\mathsf{F}(x_1) - \mathsf{F}(x_0)}{t} = \mathsf{DF}(x_0)\mathsf{F}(x_0) = \frac{\mathsf{d}}{\mathsf{d}t}\mathsf{F}(x(t))|_{t=0} = \ddot{x}(0),$$

and therefore

$$\mathsf{F}(x_1) - \mathsf{F}(x_0) = t\ddot{x}(0) + \mathcal{R}_\mathsf{F}(t\mathsf{F}(x_0)). \tag{13}$$

Next we employ (12) and (13) in order to end up with

$$x(t) - x_1 = t^2 \frac{\ddot{x}(0)}{2} + \mathcal{R}_x(t) = \frac{t}{2}(\mathsf{F}(x_1) - \mathsf{F}(x_0)) + t\mathcal{R}_\mathsf{F}(t\mathsf{F}(x_0)) + \mathcal{R}_x(t). \tag{14}$$

Remark 1 Note that (14) can serve as an error indicator for the iteration (2) (see [2, §2.3] or [3, §2.2] for further details).

Now we consider the difference $x(t) - \tilde{x}_1$ using (14):

$$\begin{aligned} x(t) - \tilde{x}_1 &= x(t) - x_1 + x_1 - \tilde{x}_1 \\ &= \frac{t}{2}(\mathsf{F}(x_1) - \mathsf{F}(x_0)) + t(\mathsf{F}(x_0) - \mathrm{proj}_v(\mathsf{F}(x_0))) + t\mathcal{R}_\mathsf{F}(t\mathsf{F}(x_0)) + \mathcal{R}_x(t) \\ &= t\left(\frac{v}{2} - \mathrm{proj}_v(\mathsf{F}(x_0))\right) + t\mathcal{R}_\mathsf{F}(t\mathsf{F}(x_0)) + \mathcal{R}_x(t). \end{aligned}$$

It is noteworthy that for $\mathsf{F}(x_1) \approx \mathsf{F}(x_0)$ there holds $\frac{v}{2} - \mathrm{proj}_v(\mathsf{F}(x_0)) \approx 0$. However, if we define $\gamma(x_0, x_1) := \left\| v/2 - \mathrm{proj}_v(\mathsf{F}(x_0)) \right\|$, we end up with the upper bound

$$\|x(t) - \tilde{x}_1\| \leq t\gamma(x_0, x_1) + \mathcal{O}(t^3). \tag{15}$$

We see that by neglecting the term $\mathcal{O}(t^3)$, the expression $t\gamma(x_0, x_1)$ can be used as an error indicator in each iteration step. Consequently, fixing a tolerance $\tau > 0$ such that

$$\tau = t\gamma(x_0, x_1), \tag{16}$$

motivates an adaptive step size control procedure for the proposed iteration scheme given in (11) that will be discussed and tested in the next section.

Adaptive Strategy

We now propose a procedure that realizes an adaptive strategy based on the previous observations. The individual computational steps are summarized in Algorithm 1.

Remark 2 By $\mathsf{R}(t)$ we signify a procedure that reduces the current step size such that $0 < \mathsf{R}(t) < t$. Let us also briefly address a possible and reasonable choice for the initial step size t_{init} in Algorithm 1. The following—detailed—reasoning can also be found in [2, §2].

If we start our procedure with a regular initial value $x_0 \in \Omega$ such that $\mathsf{F}(x) = -\mathsf{J}_f(x)^{-1} f(x)$ is Lipschitz continuous in a neighborhood of x_0, then there exists a local—unique—solution for (1), i.e., there exists $T > 0$ with

$$\dot{x}(t) = -\mathsf{J}_f(x(t))^{-1} f(x(t)),$$

on $t \in [0, T)$. Consequently, there holds $f(x(t)) = f(x_0)\mathrm{e}^{-t}$. A second order Taylor expansion reveals (see [2, §2.2] or [1, §2.2] for details)

$$x(t) \approx x_0 + \dot{x}(0)t + t^2\xi = x_0 + t\mathsf{F}(x_0) + t^2\xi, \tag{17}$$

with $\xi \in \mathbb{R}^n$ to be determined. Moreover we use a second order Taylor expansion for f and compute

$$f(x_0)\mathrm{e}^{-t} = f(x(t)) \approx f(x_0 + \dot{x}(0)t + t^2\xi) = f(x_0) - tf(x_0) + \mathsf{J}_f(x_0)t^2\xi. \tag{18}$$

We finally use $\mathrm{e}^{-t} \approx 1 - t + \frac{t^2}{2}$ in

$$f(x_0)\mathrm{e}^{-t} \approx f(x_0) - tf(x_0) + \mathsf{J}_f(x_0)t^2\xi$$

in order to end up with

$$\xi \approx \frac{1}{2}\mathsf{J}_f(x_0)^{-1} f(x_0).$$

Algorithm 1 Adaptive Newton-like method:

1: **Input**:

- an initial value $x_0 \in \Omega$,
- an initial step size t_{init}.
- a lower bound for the step size $t_{\text{lower}} > 0$,
- an error tolerance $\tau > 0$ and $\varepsilon > 0$ respectively.

2: $\mathsf{F}(x_0) \leftarrow \mathsf{A}(x_0) f(x_0)$
3: $t \leftarrow \min(1, t_{\text{init}})$
4: **for** $k = 1, 2, \ldots$ **do**
5: **if** $\|\mathsf{F}(x_0)\| \leq \varepsilon$ **then**
6: **return** $x_\infty \leftarrow x_0$
7: **else**
8: **loop** ▷ start the adaptive step size control
9: **if** $t < t_{\text{lower}}$ **then**
10: **stop the iteration procedure**
11: **end if**
12: $x_1 \leftarrow x_0 + t\mathsf{F}(x_0)$
13: $\mathsf{F}(x_1) \leftarrow \mathsf{A}(x_1) f(x_1)$
14: $v \leftarrow \mathsf{F}(x_1) + \mathsf{F}(x_0)$
15: $\text{proj}_v(\mathsf{F}(x_0)) \leftarrow \frac{\langle v, \mathsf{F}(x_0)\rangle}{\|v\|^2} v$
16: $\gamma(x_0, x_1) \leftarrow \left\| v/2 - \text{proj}_v(\mathsf{F}(x_0)) \right\|$
17: **if** $t\gamma(x_0, x_1) \leq \tau$ **then**
18: **break the loop**
19: **else**
20: $t \leftarrow \mathsf{R}(t)$ ▷ reduce the step size
21: **end if**
22: **end loop**
23: $x_0 \leftarrow x_0 + t\text{proj}_v(\mathsf{F}(x_0))$ ▷ perform a step
24: $\mathsf{F}(x_0) \leftarrow \mathsf{A}(x_0) f(x_0)$ ▷ update the direction
25: $t \leftarrow \min\left(1, \frac{\tau}{\gamma(x_0, x_1)}\right)$ ▷ predict the step size
26: **end if**
27: **end for**

Combining this with (17) yields

$$x(t) \approx x_0 + t\mathsf{F}(x_0) + \frac{1}{2}t^2 \mathsf{J}_f(x_0)^{-1} f(x_0).$$

Note that $x_1 = x_0 + t\mathsf{F}(x_0)$. Thus after a first step $t = t_{\text{init}} > 0$ we get

$$\|x(t) - x_1\| \approx \frac{1}{2}t^2 \left\| \mathsf{J}_f(x_0)^{-1} f(x_0) \right\|.$$

Thus for the given error tolerance $\tau > 0$, we set

$$t_{\text{init}} = \sqrt{\frac{2\tau}{\left\| \mathsf{J}_f(x_0)^{-1} f(x_0) \right\|}},$$

i.e., we arrive at $\|x(t_{\text{init}}) - x_1\| \approx \tau$.

Remark 3 In Algorithm 1 the minimum in Step 3 and 25 respectively is chosen such that $t = 1$ whenever possible, in particular, whenever the iterates are close to the zeros x_∞ (see Fig. 4). This will retain the famous quadratic convergence property of the classical Newton scheme (provided that the zero x_∞ is simple).

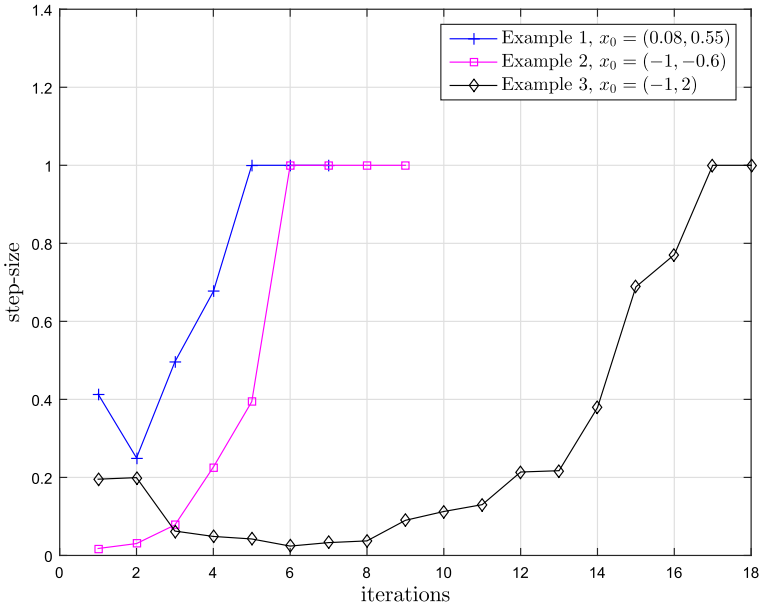

Fig. 4 Step-size versus number of effective computed updates in Algorithm 1 (Step 23). Here, x_0 denotes the initial value used in the depicted iteration (with $\tau = 0.1$ and $\varepsilon = 10^{-9}$)

Numerical Experiments

The purpose of this section is to illustrate the performance of Algorithm 1 by means of a few examples. In particular, we consider three algebraic systems. The first one is a polynomial equation on $\mathbb{C}$ (identified with $\mathbb{R}^2$) with three separated zeros, and the second example is a challenging benchmark problem in $\mathbb{R}^2$. Finally, we consider a problem in $\mathbb{R}^2$ with exactly one zero in order to highlight the fact that—in certain situations—the classical Newton method is able to find a numerical solution whereas the proposed adaptive scheme is not convergent. For all presented examples, we set in Algorithm 1

$$t_{\text{init}} = \min\left(\sqrt{\frac{2\tau}{\left\|\mathsf{J}_f(x_0)^{-1} f(x_0)\right\|}}, 1\right).$$

In Algorithm 1, the lower bound for the step size in Step 9 is set to $t_{\text{lower}} = 10^{-9}$ and for the error tolerance in Step 5 we use $\varepsilon = 10^{-8}$. Moreover, for the possible reduction procedure $t \leftarrow \mathsf{R}(t)$ in Step 20 we simply use $\mathsf{R}(t) := \frac{t}{2}$. Let us further point out that there are more sophisticated strategies for the reduction process of the time step t (see also [12, §10]). Finally, we set the maximal number of iterations $n_{\max}$ to 100.

Example 1 We consider the function

$$f : \mathbb{C} \to \mathbb{C}, \quad z \mapsto f(z) := z^3 - 1.$$

Here, we identify f in its real form in $\mathbb{R}^2$, i.e., we separate the real and imaginary parts. The three zeros are given by

$$Z_f = \{(1, 0), (-1/2, \sqrt{3}/2), (-1/2, -\sqrt{3}/2)\} \subset \mathbb{C}.$$

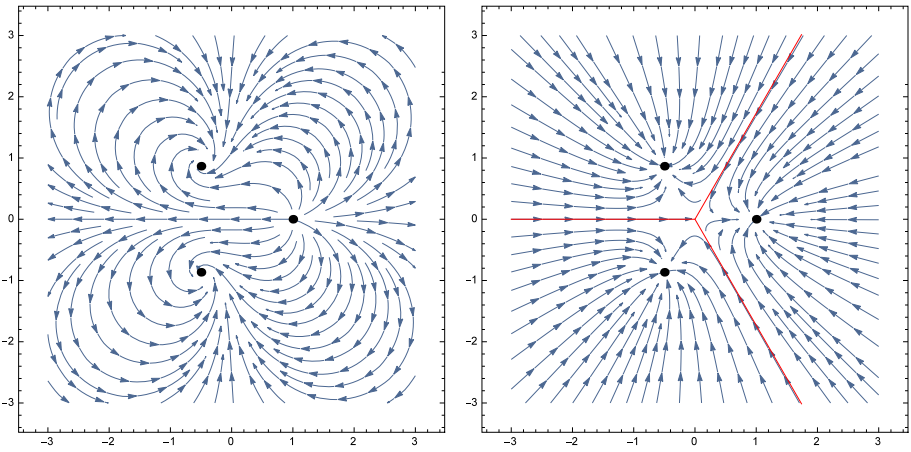

Fig. 5 Example 1: The direction fields corresponding to $f(z) = z^3 - 1$ (left) and to the transformed $\mathsf{F}(z) = -\mathsf{J}_f(z)^{-1} f(z)$ (right)

Note that J_f is singular at $(0, 0)$. Thus if we apply the classical Newton method with $\mathsf{F}(x) = -\mathsf{J}_f(x)^{-1} f(x)$ in (2), the iterates close to $(0, 0)$ causes large updates in the iteration procedure. More precisely, the application of $\mathsf{F}(x) = -\mathsf{J}_f(x)^{-1} f(x)$ is a potential source for chaos near $(0, 0)$. Before we discuss our numerical experiments, let us first consider the vector fields generated by the continuous problem (1). In Fig. 5, we depict the direction fields corresponding to $\mathsf{F}(x) = f(x)$ (left) and $\mathsf{F}(x) = -\mathsf{J}_f(x)^{-1} f(x)$ (right). We clearly see that $(1, 0) \in Z_f$ is repulsive for $\mathsf{F}(x) = f(x)$. Moreover, the zeroes $(-1/2, \sqrt{3}/2), (-1/2, -\sqrt{3}/2) \in Z_f$ of $\mathsf{F}(x) = f(x)$ show a curl. If we now consider $\mathsf{F}(x) = -\mathsf{J}_f(x)^{-1} f(x)$, the situation is completely different: All zeros are obviously attractive. In this example, we further observe that the vector direction field is divided into three different sectors, each containing exactly one element of Z_f. Next we visualize the domains of attraction of different Newton-type schemes. In doing so, we compute the zeros of f by sampling initial values on a 500×500 grid in the domain $[-3, 3]^2$ (equally spaced). In Fig. 6, we show the fractal generated by the traditional Newton method with step size $t \equiv 1$ (left) as well as the corresponding plot for the adaptive Newton-type scheme with the proposed variable step size t (right). It is noteworthy that the chaotic behavior caused by the singularities of J_f is clearly tamed by the adaptive procedure. Here, we set $\tau = 0.01$ in Algorithm 1.

Example 2 The second test example is a 2×2 algebraic system from [4] defined as

$$f : [-1.5, 1.5]^2 \to \mathbb{R}^2, \quad f(x, y) := \begin{pmatrix} \exp(x^2 + y^2) - 3 \\ x + y - \sin(3(x + y)) \end{pmatrix}. \tag{19}$$

Firstly we notice that the singular set for J_f is given by

$$\{y = x\}, \quad \text{and} \quad \left\{ y = -x \pm \frac{1}{3} \arccos\left(\frac{1}{3}\right) \pm \frac{2}{3}\pi k, k \in \mathbb{N}_{\geq 0} \right\}.$$

In Fig. 7, we again depict the direction field associated to $\mathsf{F}(x) = f(x)$ (left) and $\mathsf{F}(x) = -\mathsf{J}_f(x)^{-1} f(x)$ (right). If we use $\mathsf{F}(x) = -\mathsf{J}_f(x)^{-1} f(x)$, we clearly see that six different zeros of f are all locally attractive. The solid (red) lines in Fig. 7 (right) indicate

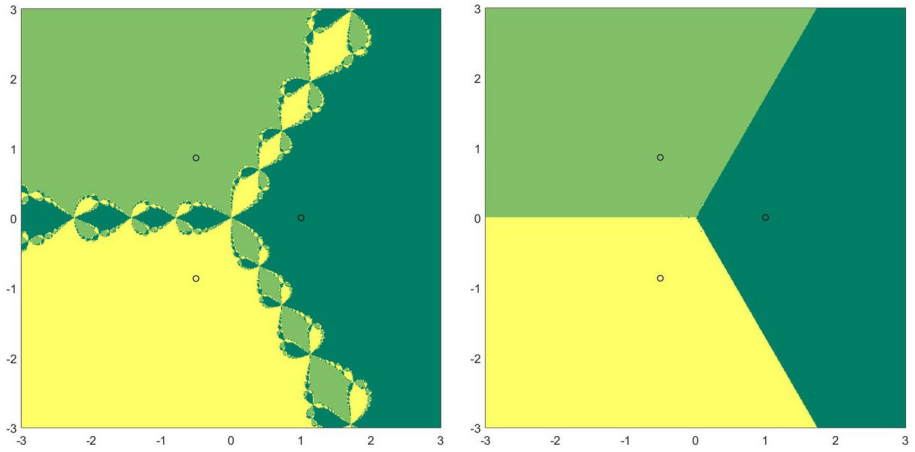

Fig. 6 The basins of attraction for Example 1 by the Newton method. On the left without step size control (i.e., $t \equiv 1$) and on the right with step size control ($\tau = 0.01$). Three different colors distinguish the three basins of attraction associated with the three solutions (each of them is marked by a small circle) (color figure online)

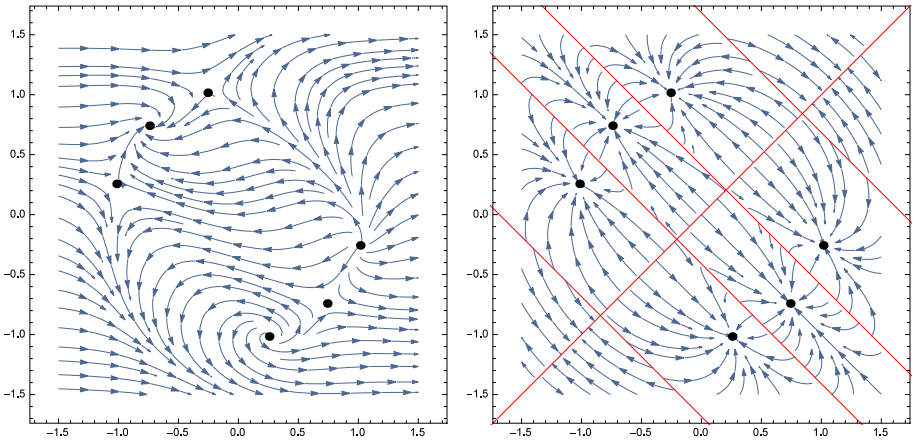

Fig. 7 Example 2: The direction fields corresponding to Example 2. On the left for $\mathsf{F}(x) = f(x)$ and to the right for the transformed vector field $\mathsf{F}(x) = -\mathsf{J}_f(x)^{-1} f(x)$

the singular set of J_f. In Fig. 8, we show the domain of attraction. We clearly see that the proposed adaptive scheme in Algorithm 1 is able to tame the chaotic behavior of the classical Newton iteration. Let us further point out the following important fact:

Suppose we are given an initial value x_0 for the continuous problem (1) which is located in the subdomain of $[-1.5, 1.5]^2$ where no root of f is located (see the upper right and the bottom left part of the domain $[-1.5, 1.5]^2$ in Fig. 7 right). The trajectories corresponding to such initial guesses end at the singular set of J_f. The situation is different in the discrete case. Indeed, if we start the Newton-type iteration in (2) on the subdomain where no zero of f is located, the discrete iteration is potentially able to cross the singular set. In addition, if we set $\tau \ll 1$, the discrete iteration (2) is close to its continuous analogue (1). Therefore a certain amount of chaos may enlarge the domain of convergence. This is particularly important when

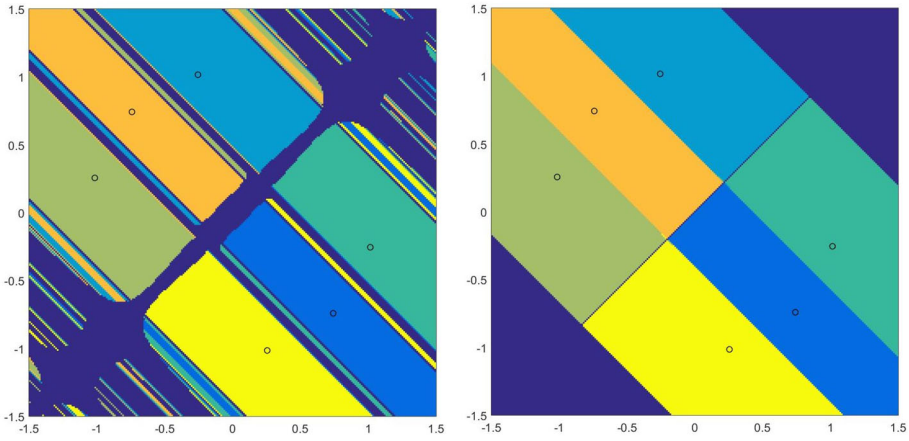

Fig. 8 The basins of attraction for Example 2 by the Newton method. On the left without step size control (i.e, $t = 1$) and on the right with step size control ($\tau = 0.01$). Six different colors distinguish the six basins of attraction associated with the six solutions (each of them is marked by a small circle). Note that the dark-blue shaded domain indicates the domain, where the iteration procedure, i.e. Algorithm 1, fails to convergence (within the maximal number of iterations which is set here to $n_{\max} = 100$) (color figure online)

no a priori information on the location of the zeros of f is available. We depict this situation in Fig. 8. Here we sample 250×250 equally spaced initial guesses on the domain $[-1.5, 1.5]^2$. The dark blue shaded part indicates the domain where the iteration fails to converge. Note that the proposed step size control is able to reduce the chaotic behavior of the classical Newton method. Moreover, the domain of convergence is again considerably enlarged by the adaptive iteration scheme.

Example 3 We finally consider the algebraic 2×2 system from [13] given by

$$f : [-10, 10]^2 \to \mathbb{R}^2, \quad f(x, y) := \begin{pmatrix} -x^2 + y + 3 \\ -xy - x + 4 \end{pmatrix}. \tag{20}$$

There exists a unique zero for f given by $(2, 1)$. This zero is an attractive fixed point for the system (1) with $\mathsf{F}(x) = f(x)$ as well as $\mathsf{F}(x) = -\mathsf{J}_f(x)^{-1} f(x)$. The associated direction fields are depicted in Fig. 9. Close to the zero $(2, 1)$ we observe a curl in case of $\mathsf{F}(x) = f(x)$. However, if we instead use $\mathsf{F}(x) = -\mathsf{J}_f(x)^{-1} f(x)$, the curl is removed and the direction field points directly to $(2, 1)$. In Fig. 10, we show the attractors of $(2, 1)$ for the classical Newton method (left) and for the proposed adaptive strategy with $\tau = 0.01$ (right). These pictures are based on sampling 10^6 starting values in the domain $[-10, 10]^2$. The right and yellow shaded part signifies the attractor for $(2, 1)$. Again we notice that the classical Newton method with step size $t \equiv 1$ produces chaos. In the adaptive case the situation is different. We clearly see that adaptivity again is able to reduce the chaos and unstable behavior of the classical Newton method. Referring to the previous Example 2, it is noteworthy that in Example 3 the domain of convergence in the adaptive case is comparable to the case of $t \equiv 1$, i.e., the classical Newton method.

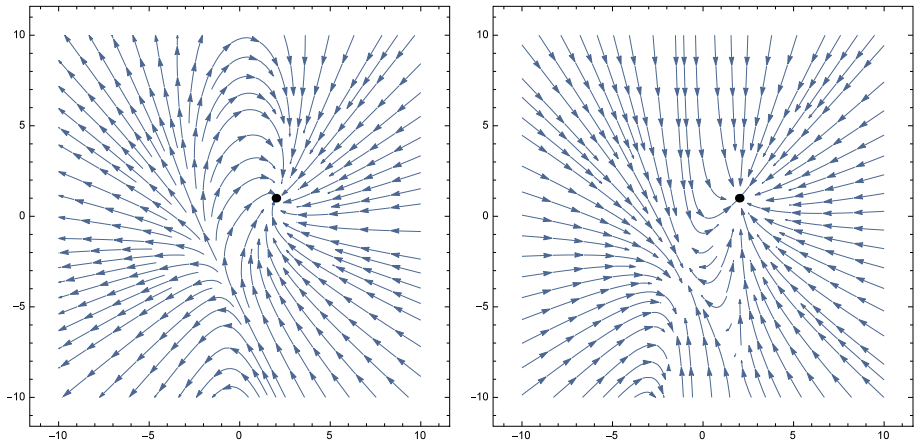

Fig. 9 The direction fields corresponding to Example 3. On the left for $\mathsf{F}(x) = f(x)$ and on the right for the transformed vector field $\mathsf{F}(x) = -\mathsf{J}_f(x)^{-1} f(x)$. We clearly see that the transformed field removes the curl which we obtain by simply applying $\mathsf{F}(x) = f(x)$

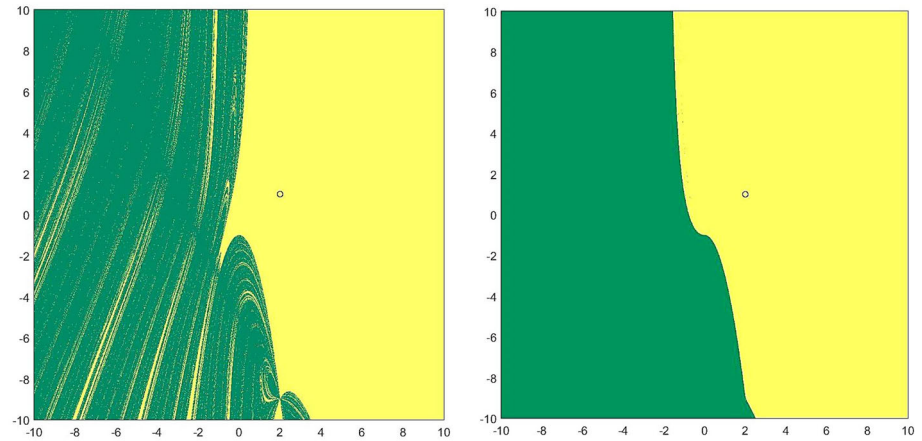

Fig. 10 The basins of attraction for Example 3 by the Newton method. On the left without step size control (i.e., $t = 1$) and on the right with step size control ($\tau = 0.01$). Note that the right part and yellow shaded domain indicates the domain, where the iteration procedure 1 converges to the unique root (2, 1) (color figure online)

Performance Data

In Fig. 11 we display the behavior of the classical and the adaptive Newton scheme (with $\tau = 0.1$). More precisely, in Example 2 we start the iteration in $x_0 = (0.08, 0.55)$. Note that x_0 is located in the exact attractor of the zero $(-1/2, \sqrt{3}/2)$. We see that the classical solution with step size $t \equiv 1$ shows large updates and thereby leaves the original attractor. On the other hand, the iterates generated by the adaptive scheme follow the exact solution (which was approximated by a numerical reference solution by choosing $t \ll 1$) quite closely and is therefore able to approach the zero which is located in the corresponding domain of attraction. In Fig. 12, we show the convergence graphs corresponding to Example 1 with

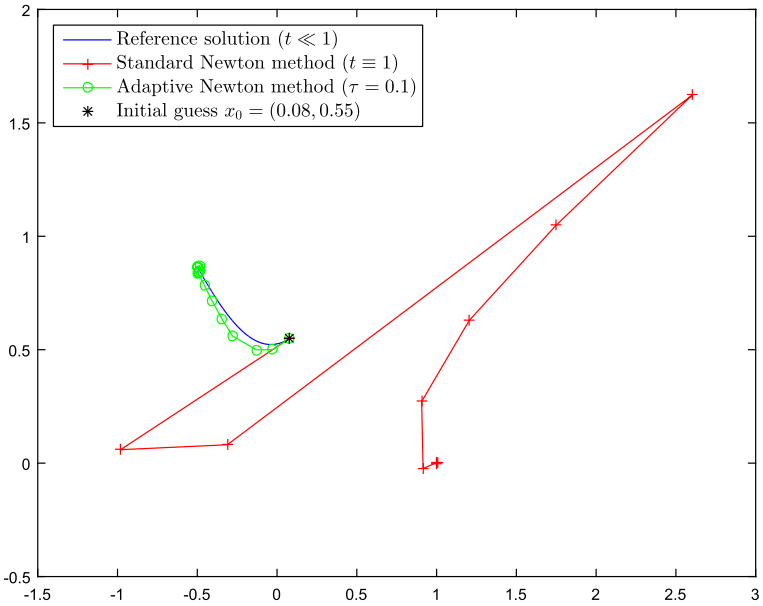

Fig. 11 Classical versus adaptive Newton method. We clearly see, that the adaptive Newton method is able to follow the reference solution leading to the *correct* zero

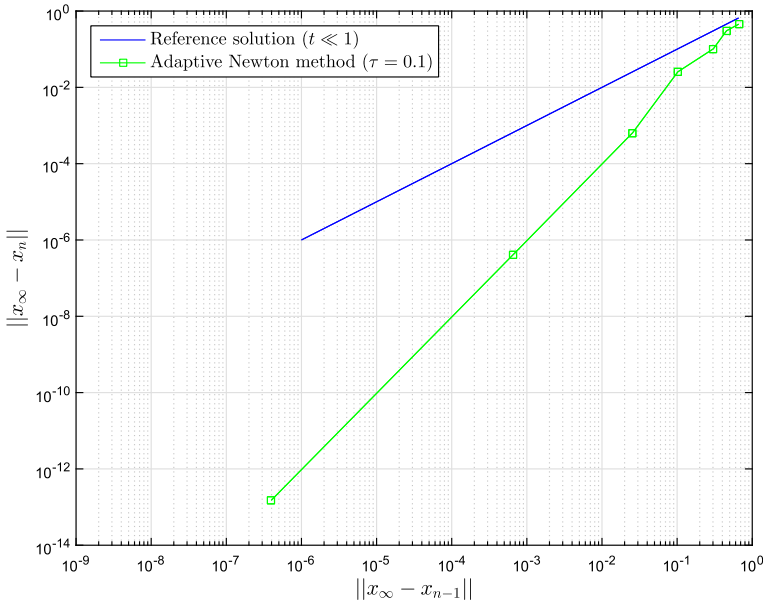

Fig. 12 The convergence graphs corresponding to the reference solution and the adaptive iteration scheme. The convergence is clearly quadratic in the adaptive iteration scheme whereas a fixed step size implies only linear convergence

Table 1 Performance for Examples 1, 2 and 3

	Example 1 on $[-3, 3]^2$ (%)	Example 2 on $[-1.5, 1.5]^2$ (%)	Example 3 on $[-10, 10]^2$ (%)
% of convergent iterations with $t \equiv 1$	88.7	55.44	51.2
% of convergent iterations with adaptive step size t	99.99	70.5	50.2

the initial guess $x_0 = (0.08, 0.55)$. Evidently, the adaptive iteration scheme shows quadratic convergence while the Newton scheme with fixed step size $t \ll 1$ converges only linearly. In Table 1, we depict the benefit of the proposed adaptive iteration scheme. The numerical results in Table 1 are based on the following considerations: For Examples 1 and 2, we sample 25×10^4 (equally-distributed) initial guesses on the domain $[-3, 3]^2$ and 2.5×10^4 on the domain $[-1.5, 1.5]^2$ respectively. Moreover, we call an initial value x_0 convergent if it is in fact convergent and, additionally, approaches the correct zero, i.e. the zero that is located in the same exact attractor as the initial value x_0. The results in Table 1 clearly show that the proposed adaptive strategy is able to enlarge the domain of convergence considerably.

Finally, let us again address Example 3 in Table 1. These results are based on the following prerequisite: Here, we call an initial value x_0 convergent if it is in fact convergent, i.e., we skip the necessity that x_0 belongs to the attractor of the unique root $x_\infty = (2, 1)$. This implies that the classical Newton method is now considered as convergent in subdomains of $[-10, 10]^2$ where the adaptive scheme is possibly not convergent (since for such an initial guess the trajectory of the continuous solution does not end at x_∞). Here, we sample 10^6 initial guesses on the domain $[-10, 10]^2$. In Table 1, we clearly see that the classical Newton method with step size $t \equiv 1$ is convergent in 51.2% of the tested values while the adaptive scheme is only convergent in 50.2% of all cases. This fact nicely demonstrates that in certain situations a chaotic behavior of the iteration process is preferable in the sense that the iterates generated by the classical scheme are possibly able to cross critical interfaces with singular Jacobian. However, —unnoticed—crossings between different basins of attraction and therefore a switching between different solutions of nonlinear problems can be considerably reduced by the proposed adaptive scheme.

Conclusions

In this work we have considered an adaptive method for Newton iteration schemes for nonlinear equations $f(x) = 0$ in $\mathbb{R}^n$. Assuming that the matrix $\mathsf{A}(x)$ is nonsingular and using $\mathsf{F}(x) = \mathsf{A}(x)f(x)$, we focus on the critical points of the ordinary differential equation $\dot{x} = \mathsf{F}(x)$. Computing the zeros of f numerically, we use an adaptive explicit iteration scheme to follow the flow generated by $\dot{x} = \mathsf{F}(x)$. Especially, since for $\mathsf{A}(x) = -\mathsf{J}_f(x)^{-1}$ the map F is nearby affine linear close to a zero x_∞, the proposed adaptivity relies on the orthogonal projection of a single iteration step onto the discretized global flow generated by the dynamics of the initial value problem $\dot{x} = \mathsf{F}(x)$. In summary, an appropriate choice of the matrix $\mathsf{A}(x)$—if possible—can lead to the favorable property of all zeros being—at least on a local level—attractive. On the other hand—especially in case of $\mathsf{A}(x) = \mathsf{J}_f(x)^{-1}$—singularities in J_f may cause the associated discrete version to exhibit chaotic behavior. In order to tame these effects, we have used an adaptive step size control procedure whose

purpose is to follow the flow of the *continuous* system to a certain extent. We have tested our method on a few low dimensional problems. Moreover, our experiments demonstrate empirically that the proposed scheme is indeed capable to tame the *chaotic* behavior of the iteration compared with the classical Newton scheme, i.e., without applying any step size control. In particular, our test examples illustrate that high convergence rates can be retained, and the domains of convergence can—typically—be considerably enlarged. It is noteworthy that the presented adaptive solution procedure is also applicable in the context of infinite dimensional problems as for example nonlinear partial differential equations. Indeed, within an adaptive finite element solution procedure the presented adaptivity in this work can serve as an adaptive step size control, thereby leading to a fully adaptive *Newton-type Galerkin* iteration scheme.

Acknowledgements Open access funding provided by ZHAW Zurich University of Applied Sciences.

References

1. Amrein, M., Wihler, T.P.: An adaptive Newton-method based on a dynamical systems approach. Commun. Nonlinear Sci. Numer. Simul. **19**(9), 2958–2973 (2014)
2. Amrein, M., Wihler, T.P.: Fully adaptive Newton–Galerkin methods for semilinear elliptic partial differential equations. SIAM J. Sci. Comput. **37**(4), A1637–A1657 (2015)
3. Amrein, M., Melenk, J.M., Wihler, T.P.: An hp-adaptive Newton–Galerkin finite element procedure for semilinear boundary value problems. Math. Methods Appl. Sci. **40**(6), 1973–1985 (2016). 13 pages, mma.4113
4. Deuflhard, P.: Newton Methods for Nonlinear Problems, Springer Series in Computational Mathematics. Springer, Berlin (2004)
5. Epureanu, B.I., Greenside, H.S.: Fractal basins of attraction associated with a damped Newton's method. SIAM Rev. **40**(1), 102–109 (1998)
6. Jacobsen, J., Lewis, O., Tennis, B.: Approximations of continuous Newton's method: an extension of Cayley's problem. In: Proceedings of the Sixth Mississippi State–UBA Conference on Differential Equations and Computational Simulations, Electronic Journal of Differential Equations Conference, vol. 15, pp. 163–173 (2007)
7. Königsberger, K.: Analysis 2, Springer-Lehrbuch, no. Bd. 2, Physica-Verlag (2006)
8. Neuberger, J.W.: Continuous Newton's method for polynomials. Math. Intell. **21**(3), 18–23 (1999)
9. Neuberger, J.W.: Integrated form of continuous Newton's method. Lect. Notes Pure Appl. Math. **234**, 331–336 (2003)
10. Neuberger, J.W.: The continuous Newton's method, inverse functions and Nash–Moser. Am. Math. Monthly **114**, 432–437 (2007)
11. Ortega, J.M., Rheinboldt, W.C.: Iterative solution of nonlinear equations in several variables. In: Classics in Applied Mathematics, vol. 30. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA, xxvi, p. 572 (2000)
12. Potschka, A.: Backward step control for global Newton-type methods. SIAM J. Numer. Anal. **54**(1), 361–387 (2016)
13. Schneebeli, H.R., Wihler, T.P.: The Newton–Raphson method and adaptive ODE solvers. Fractals **19**(1), 87–99 (2011)
14. Smale, S.: A convergent process of price adjustment and global Newton methods. J. Math. Econ. **3**(2), 107–120 (1976)

15. Tanabe, K.: Continuous Newton–Raphson method for solving an underdetermined system of nonlinear equations. Nonlinear Anal. Theory Methods Appl. **3**(4), 495–503 (1979)
16. Tanabe, K.: A geometric method in nonlinear programming. J. Optim. Appl. **30**(2), 181–210 (1980)
17. Tao, T.: The Nash–Moser iteration scheme, Technical report, tao/preprints/Expository (2006)