Applied Mathematics and Computation 320 (2018) 190–201

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Full linear multistep methods as root-finders

Bart S. van Lith[a,*], Jan H.M. ten Thije Boonkkamp[a], Wilbert L. IJzerman[a,b]

[a] Department of Mathematics and Computer Science, Eindhoven University of Technology, P. O. Box 513, Eindhoven NL-5600 MB, The Netherlands

[b] Philips Lighting – High Tech Campus 44, 5656 AE Eindhoven, The Netherlands

ARTICLE INFO

Keywords:
Root-finder
Nonlinear equation
Linear multistep methods
Iterative methods
Convergence rate

ABSTRACT

Root-finders based on full linear multistep methods (LMMs) use previous function values, derivatives and root estimates to iteratively find a root of a nonlinear function. As ODE solvers, full LMMs are typically not zero-stable. However, used as root-finders, the interpolation points are convergent so that such stability issues are circumvented. A general analysis is provided based on inverse polynomial interpolation, which is used to prove a fundamental barrier on the convergence rate of any LMM-based method. We show, using numerical examples, that full LMM-based methods perform excellently. Finally, we also provide a robust implementation based on Brent's method that is guaranteed to converge.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

Suppose we are given a sufficiently smooth nonlinear function $f : \mathbb{R} \to \mathbb{R}$ and we are asked to solve the equation

$$f(x) = 0. \tag{1}$$

This archetypical problem is ubiquitous in all fields of mathematics, science and engineering. For example, ray tracing techniques in optics and computer graphics need to accurately calculate intersection points between straight lines, rays, and objects of varying shapes and sizes [1,2]. Implicit ODE solvers are often formulated like (1), after which a root-finder of some kind is applied [3].

Depending on the properties of the function f, there are several methods that present themselves. Sometimes the derivative is not available for various reasons, in which case the secant method will prove useful. If higher-order convergence is desired, inverse quadratic interpolation may be used [4]. If the derivative of f exists and is available, Newton's method is a solid choice, especially if f is also convex.

Recently, a new interpretation of root-finding methods in terms of ODEs has been introduce by Grau-Sánchez et al. [5–7]. Their idea is to consider the inverse function derivative rule as an ODE, so that any explicit ODE solver may be converted to a root-finding method. Indeed, Grau-Sánchez et al. have successfully introduced root-finders based on Adams-type multistep and Runge–Kutta integrators. It goes without saying that only explicit ODE solvers can be usefully converted to root-finding methods. However, predictor–corrector pairs are possible, as those methods are indeed explicit.

We argue that the ODE approach can be interpreted as inverse interpolation with (higher) derivatives. Indeed, any linear integration method is based on polynomial interpolation. Thus, the ODE approach can be seen as a generalisation of inverse

* Corresponding author.
E-mail address: b.s.v.lith@tue.nl (B.S. van Lith).

https://doi.org/10.1016/j.amc.2017.09.003
0096-3003/© 2017 Elsevier Inc. All rights reserved.

interpolation methods such as the secant method or inverse quadratic interpolation. The analysis can thus be combined into a single approach based on inverse polynomial interpolation.

Our main theoretical result is a theorem on the convergence rate of root-finders based on explicit linear multistep methods. We furthermore prove a barrier on the convergence rate of LMM-based root-finders. It turns out that adding a few history points quickly boosts the convergence rate close to the theoretical bound. However, adding many history points ultimately proves an exercise in futility due to diminishing returns in the convergence rate. Two LMM-based methods are constructed explicitly, one using two history points with a convergence rate of $1+\sqrt{3}\approx 2.73$ and another with three history points that converges with rate 2.91.

In terms of the efficiency measure, defined as $p^{\frac{1}{w}}$ with p the order and w the number of evaluations [4], our method holds up even compared to several optimal memoryless root-finders. The Kung–Traub conjecture famously states the optimal order of memoryless root-finders using w evaluations is 2^{w-1}, so that the efficiency is $2^{\frac{w-1}{w}}$. Our method requires two evaluations per iteration, provided function and derivative values are stored, leading to an efficiency of 1.74 for the three-point method. As a consequence, anything up to an eighth-order method has a lower efficiency measure. For instance, optimal fourth-order methods with $\sqrt[3]{4}\approx 1.59$ [8–10], or eighth-order methods with $\sqrt[4]{8}\approx 1.68$ [11–13]. Only 16th-order methods, e.g. Geum and Kim's [14], with $\sqrt[5]{16}\approx 1.74$ would in theory be more efficient.

Using several numerical examples, we show that the LMM-based methods indeed achieve this higher convergence rate. Furthermore, pathological examples where Newton's method fails to converge are used to show increased stability. We also construct a robust LMM-based method combined with bisection to produce a method that can been seen as an extension of Brent's [15]. Similar to Brent's method, whenever an enclosing starting bracket is provided, an interval $[a, b]$ with $f(a)f(b)<0$, our method is guaranteed to converge.

This article is organised in the following way. First, we find the convergence rate of a wide class of root-finders in Section 2 and prove a barrier on the convergence rates. Next, in Section 3 we derive new root-finders based on full linear multistep methods and show that such methods are stable when the initial guess is sufficiently close to the root. After this, some results are presented in Section 4 that verify our earlier theoretical treatment. Finally, we present our robust implementation in Section 5, after which we give our conclusions in Section 6.

2. Barriers on LMM root-finders

Root-finding methods based on the ODE approach of Grau-Sánchez et al. can be derived by assuming that the function f is sufficiently smooth and invertible in the vicinity of the root. Under these assumptions, the chain rule gives

$$\frac{\mathrm{d}x}{\mathrm{d}y}=[f^{-1}]'(y)=\frac{1}{f'\big(x\big)}=F(x), \tag{2}$$

which we may interpret as an autonomous ODE for the inverse. Integrating (2) from an initial guess $y_0=f(x_0)$ to $y=0$ yields

$$x(0)=x_0+\int_{y_0}^{0}F\big(x(y)\big)\mathrm{d}y, \tag{3}$$

where $x(0)$ is the location of the root. Immediately, we see that applying the forward Euler method to (2) gives Newton's method. From (3), we see that the step size of the integrator should be taken as $0-y_0=-f(x_0)$. However, Newton's method may also be interpreted as an inverse linear Taylor method, i.e., a method where the inverse function is approximated by a first-order Taylor polynomial. Indeed, any linear numerical integration method applied to (2) can be interpreted as an inverse polynomial interpolation.

As such, explicit linear multistep methods applied to (2) will also produce a polynomial approximation to the inverse function. Such a method has the form

$$x_{n+s}+\sum_{k=0}^{s-1}a_k^{(n)}x_{n+k}=h_{n+s}\sum_{k=0}^{s-1}b_k^{(n)}F(x_{n+k}), \tag{4}$$

where indeed $b_s^{(n)}=0$, otherwise we end up with an implicit root-finder, which would not be very useful. The coefficients of the method, $\{a_k^{(n)}\}_{k=0}^{s-1}$ and $\{b_k^{(n)}\}_{k=0}^{s-1}$, will depend on the previous step sizes and will therefore be different each step. The step sizes are given by $h_{n+k}=y_{n+k}-y_{n+k-1}$, the differences in y-values. Since we wish to find the root, we set $y_{n+s}=0$, leading to $h_{n+s}=y_{n+s}-y_{n+s-1}=-y_{n+s-1}$. Furthermore, the y-values are of course given by the function values of the root estimates, i.e.,

$$h_{n+k}=f(x_{n+k})-f(x_{n+k-1})\quad\text{for } k=1,\dots,s-1. \tag{5}$$

Like an ODE solver, we may use an implicit LMM in tandem with an explicit LMM to form a predictor–corrector pair, the whole forming an explicit method. Unlike an ODE solver, we may construct derivative-free root-finders based on the LMM approach by setting all $b_k^{(n)}=0$ for $k=0,\dots s-1$ and for all $n>0$, e.g., the secant method. For an ODE solver this would

Download English Version:

https://daneshyari.com/en/article/5775466

Download Persian Version:

https://daneshyari.com/article/5775466

Daneshyari.com