Applied Mathematics and Computation 295 (2017) 95–113

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Symbolic computation and computer graphics as tools for developing and studying new root-finding methods

I. Petković[a,*], Đ. Herceg[b]

[a] Faculty of Electronic Engineering, University of Niš, A. Medvedeva 14, Niš 18000, Serbia
[b] Department of Mathematics and Informatics, Faculty of Science, Trg Dositeja Obradovića 4, University of Novi Sad, Novi Sad 21 000, Serbia

ARTICLE INFO

Keywords:
Symbolic computation
Solving nonlinear equations
Multiple zeros
Computer graphics
Dynamic study
Basin of attraction

ABSTRACT

Many very difficult problems in applied mathematics and other scientific disciplines cannot be solved without powerful computational systems, such as symbolic computation and computer graphics. In this paper we construct two new families of the fourth order iterative methods for finding a multiple real or complex zero of a given function. For developing these methods, a recurrent formula for generating iterative methods of higher order for solving nonlinear equations is applied and implemented by symbolic computation through several programs in computer algebra system *Mathematica*. Symbolic computation was the only tool for solving the considered complex problem since it provides handling and manipulating complex mathematical expressions and other mathematical objects. The properties of the proposed rapidly convergent methods are illustrated by several numerical examples. To examine the convergence behavior of the presented methods, we also give the dynamic study of these methods using basins of attraction. Such a methodology, besides a visualization of iterative processes, deliveries very important features on iterations including running CPU time and average number of iterations, as a function of starting points. The program for plotting basins of attraction in *Mathematica* is included.

1. Introduction

In many scientific disciplines such as applied mathematics, engineering disciplines, physics, chemistry, computer science, biology, astronomy, geology, as well as many other fields of human activities, the solution to a lot of complex problems had to wait for the development of computer hardware and software to reach a more advanced level. Many difficult problems remained unsolved due to the lack of powerful hardware and advanced and sophisticated software. As commented in [1], "at the beginning of the 21st century the rapid development of computer power and accessibility, computer multi-precision arithmetics and symbolic computation enabled the construction, testing and analysis of very efficient numerical algorithms, even confirmation of analytically derived results." Symbolic computation has successfully substituted lengthy manual calculation with computer-based computation and manipulation. In particular, in numerical mathematics further development of new algorithms of higher efficiency and higher accuracy has become possible. In this paper we concentrate on methods and procedures for the construction, analysis and practical application of algorithms for solving nonlinear equations with the support of symbolic computation. We emphasize that the construction of presented root-solvers would be hardly feasible and most likely impossible without the use of this specific computer software.

* Corresponding author.
E-mail addresses: ivan.petkovic@elfak.ni.ac.rs, ivan.petkovic@gmail.com (I. Petković), herceg@dmi.uns.ac.rs (Đ. Herceg).

http://dx.doi.org/10.1016/j.amc.2016.09.025
0096-3003/© 2016 Elsevier Inc. All rights reserved.

Symbolic computation (often called *computer algebra systems*) for solving mathematical problems deals with mathematical expressions and other mathematical objects, even when these expressions contain variables in a general form such as $f(a, b, c, \ldots)$, where a, b, c are not numerical values and f need not be given explicitly. Among many manipulations, this specific software enables automatic simplification of expressions, differentiation, integration, polynomial manipulations and even exact computation, treating the mentioned variables as symbols. The interested reader is referred to the books by Cohen [2] and Beiley et al. [3] for more details on symbolic computation and its applications.

Another tool for comparison and analysis of root-finding algorithms, presented in this paper, belongs to the area of computer graphics. It provides a deep and fruitful insight into visualization of approximating function zeros using basins of attraction, see, e.g. [4] and [5]. Such approach dramatically improves the quality estimation of iterative methods concerning areas of convergence. Combined with the study of computational efficiency of root-finding methods, it leads to considerably better understanding of iterative processes.

Today, mathematicians and computer scientists carry out sophisticated mathematical operations employing powerful computer machines supported by the modern computer algebra systems such as Mathematica, Maple, Axiom, GAP, Maxima, Sage and SymPy. These computer algebra systems, which enable both symbolic computation and a dynamic study using basins of attraction, are available on Windows, Mac OS X and Linux.

There is a vast number of papers devoted to iterative methods for finding multiple zeros, see, e.g., [6–17]. In this paper we are concerned with generating iterative methods of higher order for finding approximations to multiple roots of nonlinear equations. For this purpose, we use a suitable generating recurrence formula for the construction of an iterative method of order $n+1$ starting from a known iterative method of order n. In this process there appear the derivative of complicated and lengthy mathematical expressions and necessary simplifications. For this reason, we were forced to employ symbolic computation through several programs written in computer algebra system *Mathematica*. Our attention was restricted to the two families of iterative processes for approximating multiple zeros: Laguerre's type methods and Traub–Gander's methods, both of the cubic convergence. The application of the mentioned generating recurrence relation to these families gives two new families of order four. In addition, in Section 3 it is shown that there is a connection between the Laguerre family and the Traub–Gander family. In this way we derive some new fourth order methods for multiple zeros.

Convergence properties of the derived methods are demonstrated in Section 4 by four numerical examples. In Section 5 we use basins of attraction to study and compare the presented iterative methods as a function of starting points. Besides useful data about CPU time and average number of iterations for 360,000 points belonging to a square, a visual insight into convergence behavior of the tested methods was enabled in this way. There is also an Appendix which contains a complete program written in *Mathematica* for plotting basins of attraction and providing useful information on iteration process. We consider that this program will be of benefit to the readers since complete programs written in *Mathematica* are very seldom presented in the literature.

2. Laguerre-like methods

Let f be a given sufficiently many times differentiable function in some complex domain S with a simple or multiple zero α. One of the best known iterative methods for finding approximations to α is the third order Laguerre family of iterative methods

$$\hat{z} = z - \frac{\nu f(z)}{f'(z) \pm \sqrt{(\nu-1)^2 f'(z)^2 - \nu(\nu-1) f(z) f''(z)}}, \tag{1}$$

where $\nu\,(\neq 0, 1)$ is a parameter and $\hat{z}$ is a new approximation. There are a lot of papers and book chapters devoted to Laguerre's method (1), see for instance [15–24]. This method possesses very good convergence properties and robustness so that it is often used in modern software packages. Note that Laguerre's method shows extremely good behavior when $|z|$ is large, see Parlett [22].

The main goal of this section is to construct Laguerre-like methods of fourth order for finding multiple zeros. We start with the family of third order iterative methods of Laguerre's type for finding a multiple zero α of f of the known multiplicity m, presented in the form

$$L_m(f, \nu; z) := z - \frac{m\nu f(z)/f'(z)}{1 + s\sqrt{(\nu-1)\left[m\nu - 1 - m\nu \dfrac{f(z) f''(z)}{f'(z)^2}\right]}} \quad (\nu \neq 0, 1), \tag{2}$$

which was known to Bodewig [19]. The symbol $s \in \{-1, 1\}$ denotes the sign which should be chosen in front of the square root.

The following alternative form of (2) (setting $m\nu = \lambda$ in (2)) has been presented in [16]

$$L_m(f, \lambda; z) := z - \frac{\lambda f(z)/f'(z)}{1 + \operatorname{sgn}(\lambda - m)\sqrt{\left(\dfrac{\lambda - m}{m}\right)\left[\lambda - 1 - \lambda \dfrac{f(z) f''(z)}{f'(z)^2}\right]}} \quad (\lambda \neq 0, m). \tag{3}$$