Intern. J. Computer Math. 1979, Section B, Vol. 7, pp. 157–161

A Sixth-Order Family of Methods for Nonlinear Equations

BENY NETA

Northern Illinois University, Department of Mathematical Sciences, DeKalb, Illinois 60115, U.S.A.

A one-parameter family of sixth-order methods for finding simple zeros of nonlinear functions is developed. Each member of the family requires three evaluations of the given function and only one evaluation of the derivative per step.

1. INTRODUCTION

Newton's method for computing a simple zero ξ of a nonlinear equation $f(x)=0$ has been modified in a number of ways. For example, Ostrowski [1] discusses a third-order method that evaluates the function f at every substep but only requires the derivative f' at every other substep. He also introduced a fourth-order scheme that uses the same information. King [2] has shown that there is a family of such methods. Traub [3] introduced a third-order method which requires one function and two derivative evaluation per step. Jarratt [4] developed fourth-order method which uses the same information. King [5] developed a fifth-order scheme that requires two evaluations of f and f'.

Here we develop methods of order six. An iteration consists of one Newton substep followed by two substeps of "modified" Newton (i.e., using the derivative of f at the first substep instead of the current one).

Let us recall the definition of order (see e.g. [3]).

DEFINITION Let $x_1, x_2, \ldots, x_i$, be a sequence converging to ξ. Let

$$\varepsilon_i = x_i - \xi.$$

If there exists a real number p and a nonzero constant C such that

$$\frac{|\varepsilon_{i+1}|}{|\varepsilon_i|^p} \to C \tag{2}$$

then p is called the order of the sequence.

2. DEVELOPMENT OF THE SIXTH-ORDER FAMILY

Development of the sixth-order family.

Let

$$\left.\begin{aligned} \omega_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ z_n &= \omega_n - \frac{f(\omega_n)}{f'(x_n)} \cdot \frac{f(x_n)+Af(\omega_n)}{f(x_n)+Bf(\omega_n)} \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(x_n)} \cdot \frac{f(x_n)+Cf(\omega_n)+Df(z_n)}{f(x_n)+Gf(\omega_n)+Hf(z_n)} \end{aligned}\right\} \tag{3}$$

where A, B, C, D, G, H are arbitrary constants. This is a family of methods which uses Newton's method in the first substep and two Newton-like in the other two substeps. In each step we have to evaluate the function $f(x)$ at the three points x_n, ω_n, z_n and to evaluate the derivative at one point x_n.

In order to find the order of the method we used MACSYMA (Project MAC's SYmbolic MAnipulation system, which is a large computer programming system written in LISP and used for performing symbolic as well as numerical mathematical manipulations [6]).

The error expression at ω_n is given by

$$\begin{aligned} \varepsilon(\omega_n) = {} & \tfrac{1}{2}F_2\varepsilon_n^2 + \tfrac{1}{6}(2F_3 - 3F_2^2)\varepsilon_n^3 + \tfrac{1}{24}(-14F_2F_3 + 12F_2^3 + 3F_4)\varepsilon_n^4 \\ & + \tfrac{1}{120}(-20F_2^2 + 100F_2^2F_3 - 25F_2F_4 - 60F_2^4 + 4F_5)\varepsilon_n^5 \\ & + \tfrac{1}{720}(5F_6 - 39F_5F_2 - 85F_4F_3 + 210F_4F_2^2 + 330F_2F_3^2 - 780F_3F_2^3 \\ & + 360F_2^5)\varepsilon_n^6 + \ldots \end{aligned} \tag{4}$$

where

$$F_i = \frac{f^{(i)}(\xi)}{f'(\xi)} \tag{5}$$

and

$$\varepsilon_n = \varepsilon(x_n) = x_n - \xi. \tag{6}$$

For later use we also give the expression for

$$\begin{aligned} f(\omega_n) = {} & \{\tfrac{1}{2}F_2\varepsilon_n^2 + \tfrac{1}{6}(F_3F_1 - 3F_2^3)\varepsilon_n^3 + \tfrac{1}{24}(3F_4 - 14F_2F_3 + 15F_2^3)\varepsilon_n^4 \\ & + \tfrac{1}{120}(4F_5 - 25F_2F_4 - 20F_3^2 + 120F_2^2F_3 - 90F_2^4)\varepsilon_n^5 \end{aligned}$$

$$+\tfrac{1}{720}(5F_6-39F_5F_2-85F_4F_3+255F_4F_2^2+370F_3^2F_2$$
$$-1095F_3F_2^3+630F_2^5)\varepsilon_4^6+\ldots\}f'(\xi) \tag{7}$$

and

$$f'(x_n)[f(x_n)+Bf(\omega_n)]=f'^2(\xi)\{\varepsilon_n+\tfrac{1}{2}(B+3)F_2\varepsilon_n^2$$
$$+\tfrac{1}{6}[(2B+4)F_3+3F_2^2]\varepsilon_n^3$$
$$+\tfrac{1}{24}[(3B+5)F_4+10F_2F_3+3BF_2^3]\varepsilon_n^4$$
$$+\tfrac{1}{120}[(4B+6)F_5+15F_2F_4+10F_3^2+20BF_2^2F_3$$
$$-15BF_2^4]\varepsilon_4^5$$
$$+\tfrac{1}{720}[(5B+7)F_6+21F_5F_2+35F_4F_3+45BF_4F_2^2$$
$$+40BF_2F_3^2-150BF_3F_2^3+90BF_2^5]\varepsilon_n^6+\ldots \tag{8}$$

The error expression at the point z_n is given by

$$\varepsilon(z_n)=\tfrac{1}{4}(B-A+2)F_2^2\varepsilon_n^3+\tfrac{1}{24}[(8B-8A+14)F_2F_3+(-3B^2$$
$$+(3A-21)B+21A-27)F_2^3]\varepsilon_n^4+\tfrac{1}{144}[(18B-18A+30)F_2F_4$$
$$+(16B-16A+24)F_3^2+(-36B^2+(36A-228)B+228A$$
$$-264)F_2^2F_3+(9B^3+90B^2-9AB^2-90AB+297B-297A$$
$$+270)F_2^4]\varepsilon_n^5+\tfrac{1}{1440}[(48B-48A+78)F_2F_5+(120B-12A$$
$$+170)F_3F_4+(-135B+135AB-825B+825A-930)F_2^2F_4$$
$$+(-240B^2+240AB-1360B+1360A-1400)F_2F_3^2+(240B^3$$
$$-240AB^2+2220B^2-2220AB+6750B-6750A+5640)F_2^3F_3$$
$$+(-45B^4+45AB^3-585B^3+585AB^2-2790B^2+2790AB$$
$$-5805B+5805A-3960)F_2^5]\varepsilon_n^6+\ldots\} \tag{9}$$

In order to annihilate the coefficient of ε_n^3 in the expression for $\varepsilon(z_n)$ we have chosen $B=A-2$. We have also chosen $c=-1$, $G=-3$ and $H=D$ so that the low-order terms in the expression for ε_{n+1} will vanish. This choice leads to the family

$$\left.\begin{aligned} \omega_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ z_n &= \omega_n - \frac{f(\omega_n)}{f'(x_n)} \frac{f(x_n) + Af(\omega_n)}{f(x_n) + (A-2)f(\omega_n)} \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(x_n)} \frac{f(x_n) - f(\omega_n) + Df(z_n)}{f(x_n) - 3f(\omega_n) + Df(z_n)} \end{aligned}\right\} \tag{10}$$

with error term

$$\varepsilon_{n+1} = \tfrac{1}{144}[2F_3^2F_2 - 3(2A+1)F_2^3F_3]\varepsilon_n^6 + \dots \tag{11}$$

Note that the error term does not depend on D so we can let $D=0$.

If the parameter A is chosen so that $A = -\frac{1}{2}$ then the $F_2^3F_3$ term in error is eliminated so that

$$\left.\begin{aligned} \omega_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ z_n &= \omega_n - \frac{f(\omega_n)}{f'(x_n)} \frac{f(x_n) - \frac{1}{2}f(\omega_n)}{f(x_n) - \frac{5}{2}f(\omega_n)} \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(x_n)} \frac{f(x_n) - f(\omega_n)}{f(x_n) - 3f(\omega_n)} \end{aligned}\right\} \tag{12}$$

and

$$\varepsilon_{n+1} = \tfrac{1}{72}F_3^2F_2\varepsilon_n^6 + \dots \tag{13}$$

3. NUMERICAL EXAMPLES

Computer tests for the functions $f = x^n - 1$ with the root $\xi = 1$ for various values of the integer n show that the method is indeed of order 6.

Remark Our method needs the same number of function evaluations as a method constructed from one substep of Newton's method followed by two substeps of the secant method. The order of the secant method is approximately 1.62 (see e.g. [7]) thus such a scheme will approximately be of order 5.2 (see e.g. [4]).

References

[1] A. M. Ostrowski, *Solution of Equations and Systems of Equations*, Academic Press, New York, 1960.

[2] Richard F. King, A family of fourth-order methods for nonlinear equations, submitted to *Siam J. Numer. Anal.*

[3] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

[4] P. Jarratt, Some efficient fourth-order multipoint methods for solving equations, *BIT* **9** (1969), 119–124.

[5] Richard F. King, A fifth-order family of modified Newton methods, *BIT* **11** (1971), 409–412.

[6] R. Bogen *et al.*, *MACSYMA Reference Manual*, MIT, Cambridge, Mass., 1975.

[7] D. M. Young and R. T. Gregory, *A Survey of Numerical Mathematics*, Addison-Wesley, Reading, Mass., 1973.

Intern. J. Computer Maths. 1979, Vol. 7, pp. 162–163

Printed in Great Britain

Book Review

THE MYTHICAL MAN-MONTH: ESSAYS ON SOFTWARE ENGINEERING, by Frederick P. Brooks, Jr. Addison-Wesley Publishing Co., Reading, Massachusetts, 1975.

This is, as the title implies, a collection of essays on topics in the area of software engineering. As the former manager of Operating System/360, Brooks has experience and a personal viewpoint worth listening to. Because of his involvement in OS/360 and its difficulties he came to ask the question: why is programming hard to manage? This book is his answer. It is directed primarily at the professional manager of programmers and professional programmers.

As Brooks tells us, Chapters 2–7 contain the central argument, "large programming projects suffer management problems different in kind from small ones, due to division of labor".

Chapter 1 considers the vast difference between a programming systems *product* and a program; the joys of programming and its woes.

Chapter 2 demythologizes the "man-month" as a unit of work, leading to Brooks Law: Adding manpower to a late software project makes it later.

Chapter 3 discusses the chief programmer team concept and argues cogently for its use in very large programming projects. His principal argument is the minimization of intercommunication within the overall group. For 20 CPTs there would be intercommunication between the 20 chief programmers rather than between the 20 group members.

Chapters 4 and 5 deal with system design. The product needs to have conceptual integrity. It should reflect a single philosophy and the user specification should flow from a few minds (too many cooks spoil the broth). Secondly, the architect of the system should have sufficient experience to overcome the natural tendency to add lots of extras to a basic design.

Chapters 6 and 7 deal with communication, communication of the user specification and design to the programmers and communication among the project team members.

The other chapters deal with scheduling and keeping to it; reasons for documenting decisions and programs; pilot projects; programming tools;