Available at WWW.MATHEMATICSWEB.ORG
POWERED BY SCIENCE @ DIRECT•

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

ELSEVIER Journal of Computational and Applied Mathematics 157 (2003) 227–230

www.elsevier.com/locate/cam

Letter to the Editor

A modified Newton method for rootfinding with cubic convergence

H.H.H. Homeier[a,b,*]

[a]Institut für Physikalische und Theoretische Chemie, Universität Regensburg, 93040 Regenburg, Germany
[b]science + computing ag, Ingolstädter Str. 22, 80807 München, Germany

Received 9 January 2003

Abstract

We consider a modification of the Newton method for finding a zero of a univariate function. The case of multiple roots is not treated. It is proven that the modification converges cubically. Per iteration it requires one evaluation of the function and two evaluations of its derivative. Thus, the modification is suitable if the calculation of the derivative has a similar or lower cost than that of the function itself. Classes of such functions are sketched and a numerical example is given.

MSC: 41A25; 65D99

Keywords: Rootfinding; Newton method

The Newton method for solving the equation $f(x_*) = 0$ uses the iteration function

$$\Phi_f(x) = x - \frac{f(x)}{f'(x)}, \tag{1}$$

and thus, an iteration sequence $x_{n+1} = \Phi_f(x_n)$ is obtained for some starting value x_0. Since $f(x_*) = 0$ entails $\Phi_f'(x_*) = 0$, the convergence to the fixed point x_* of Φ_f, i.e., to the root x_* of f is locally quadratic if $f'(x_*) \neq 0$. In case that the root is m-fold, one may employ the modified Newton-type

* Corresponding author. Institute of Theoretical chemistry, University of Regensburg, Universitaetsstr. 31, 93040 Regensburg, Germany. Tel.: +491716290224; fax: +498935638637.
E-mail address: herbert.homeier@na-net.ornl.gov (H.H.H. Homeier).

0377-0427/03/$ - see front matter © 2003 Published by Elsevier B.V.
doi:10.1016/S0377-0427(03)00391-1

iteration function

$$\Phi_{f,m}(x) = x - m\,\frac{f(x)}{f'(x)}, \tag{2}$$

that again leads to quadratical convergence.

In the sequel, we consider only the case $f'(x_*) \neq 0$. Thus, modifications of the Newton method for multiple roots will be considered elsewhere.

Here, we study iteration functions of the type

$$\Psi_f(x) = x - \frac{f(x)}{f'(x + a(x)f(x))}. \tag{3}$$

Simple manipulations yield

$$\Psi_f'(x) = 1 - \frac{f'(x)}{f'(x + a(x)f(x))} + \frac{f(x)f''(x + a(x)f(x))(1 + a'(x)f(x) + a(x)f'(x))}{(f'(x + a(x)f(x)))^2} \tag{4}$$

and

$$\begin{aligned}
\Psi_f''(x) = &-\frac{f''(x)}{f'(x + a(x)f(x))}\\
&+\frac{2f'(x)f''(x + a(x)f(x))(1 + a'(x)f(x) + a(x)f'(x))}{(f'(x + a(x)f(x)))^2}\\
&-\frac{2f(x)(f''(x + a(x)f(x)))^2(1 + a'(x)f(x) + a(x)f'(x))^2}{(f'(x + a(x)f(x)))^3}\\
&+\frac{f(x)f'''(x + a(x)f(x))(1 + a'(x)f(x) + a(x)f'(x))^2}{(f'(x + a(x)f(x)))^2}\\
&+\frac{(f(x))^2 f''(x + a(x)f(x))a''(x)}{(f'(x + a(x)f(x)))^2}\\
&+\frac{f(x)f''(x + a(x)f(x))(2a'(x)f'(x) + a(x)f''(x))}{(f'(x + a(x)f(x)))^2}.
\end{aligned} \tag{5}$$

Hence, for $x = x_*$ we obtain using $f(x_*) = 0$

$$\Psi_f(x_*) = x_*, \tag{6}$$

which means that the root is also a fixed point of Ψ_f, and

$$\Psi_f'(x_*) = 0. \tag{7}$$

The latter also implies that for the iterative scheme $x_{n+1} = \Psi_f(x_n)$, quadratic convergence is obtained for starting values x_0 sufficiently close to x_* for any smooth function a. Further, we obtain

$$\begin{aligned}
\Psi_f''(x_*) &= -\frac{f''(x_*)}{f'(x_*)} + \frac{2f'(x_*)f''(x_*)(1 + a(x_*)f'(x_*))}{(f'(x_*))^2}\\
&= \frac{f''(x_*)}{f'(x_*)}(1 + 2a(x_*)f'(x_*)).
\end{aligned} \tag{8}$$

This means that for $a(x_*) = -[2f'(x_*)]^{-1}$ even $\Psi_f''(x_*) = 0$ is obtained. The easiest way to enforce this condition is to put $a(x) = -[2f'(x)]^{-1}$. But other choices of a are equally compatible with the above condition, for instance $a(x) = -[2f'(x)]^{-1} + b(x)f(x)$. The latter will not be considered in the sequel. The conditions $\Psi_f(x_*) = x_*$, $\Psi_f'(x_*) = 0$, and $\Psi_f''(x_*) = 0$ imply locally cubic convergence of the iterative scheme $x_{n+1} = \Psi_f(x_n)$. This follows for sufficiently smooth iteration functions Ψ_f using the Taylor expansion around $x = x_*$:

For $e_n = x_n - x_*$ we obtain

$$e_{n+1} = e_n \Psi_f'(x_*) + \frac{e_n^2}{2!} \Psi_f''(x_*) + \frac{e_n^3}{3!} \Psi_f'''(x_* + \Theta_n e_n) \tag{9}$$

with $0 \leqslant \Theta_n \leqslant 1$. If there is some constant M with $|\Psi_f'''(x)| \leqslant M$ in some neighborhood of x_*, the estimate

$$|x_{n+1} - x_*| \leqslant |x_n - x_*|^3 M/3! \tag{10}$$

characteristic of locally cubic convergence holds in this neighborhood.

Thus, we have proved the following theorem:

Theorem 1. *Assume that the function f is sufficiently smooth in a neighborhood of its root x_* where $f'(x_*) \neq 0$, that the iteration function $\Psi_f : x \to x - f(x)/f'(x - f(x)/(2f'(x)))$ satisfies $|\Psi_f'''(x)| \leqslant M$ for some constant M in that neighborhood. Then the iterative scheme $x_{n+1} = \Psi_f(x_n)$ converges cubically to x_* in a neighborhood of x_*.*

Obviously, the choice $a(x) = -[2f'(x)]^{-1}$ implies that for each calculation of $\Psi_f(x)$ one has to evaluate the function f once and the derivative function f' twice. Thus, the proposed iterative algorithm is particularly suited for finding zeroes of functions for which the derivative is easy to calculate. One class of these are polynomial functions f. Another important class are functions defined via integrals like $f : x \to \int_a^x F(t)\,dt + G(x)$. In fact, for such functions, the numerical evaluation of the derivative is much easier than that of the function itself if G is a constant, or a polynomial, say. A numerical example is presented in the following.

Consider the function

$$f : x \to \int_0^x [\exp(-t^3/2) - \exp(-t^8/2)]\,dt + 0.1 \tag{11}$$

with derivative

$$f' : x \to \exp(-x^3/2) - \exp(-x^8/2) \tag{12}$$

for real x. The function f has a zero close to -0.8805978315532975. The calculations were done using MAPLE V™ Release 5 [1,2] on a personal computer requiring 16 decimal digits for all numerical evaluations. The function f was evaluated using the standard numerical quadrature scheme in MAPLE V™ Release 5 that is a based on adaptive combination of Clenshaw–Curtis rules with double-exponential methods and singularity detection.

The performance of the present method with a standard Newton method is compared in Table 1. It is easily seen that much fewer iterations are used by the modified Newton method due to its higher order. In addition, the x_n are closer to the zero on average. This leads to the improved calculation time since closer inspection of the used methods reveals that for $x < -1.29$,

Table 1
Comparison of the Newton method with the modified Newton method

(n)	Mod. Newton		Newton	
	x_n	t_n	x_n	t_n
1	−0.4707395081663049	0.149	−2.446862619356371	0.144
2	−0.4999786132893553	0.071	−2.321648431610980	2.535
3	−0.5417436071987847	0.154	−2.179178506600311	1.549
4	−0.6082138921935461	0.150	−2.012613420847618	1.941
5	−0.7208001410567703	0.080	−1.812019928261384	1.995
6	−0.8484610468432506	0.155	−1.567523037512359	1.252
7	−0.8800872980821578	0.069	−1.291251881924022	0.197
8	−0.8805978314985499	0.154	−1.067133257270631	0.075
9	−0.8805978315532975	0.070	−0.9419145648006518	0.156
10	−0.8805978315532975	0.160	−0.8892819901342697	0.071
11	−0.8805978315532975	0.000	−0.8807923930637992	0.154
12	−0.8805978315532975	0.000	−0.8805979309632560	0.151
13	−0.8805978315532975	0.000	−0.8805978315533234	0.079
14	−0.8805978315532975	0.000	−0.8805978315532975	0.155
15	−0.8805978315532975	0.000	−0.8805978315532975	0.070
16	−0.8805978315532975	0.000	−0.8805978315532975	0.000
17	−0.8805978315532975	0.000	−0.8805978315532975	0.000
18	−0.8805978315532975	0.000	−0.8805978315532975	0.000
19	−0.8805978315532975	0.000	−0.8805978315532975	0.000
20	−0.8805978315532975	0.000	−0.8805978315532975	0.000
	$T_{20} = 1.217$		$T_{20} = 10.530$	

For both methods the x_n for $x_0 = -0.45$ and the computation times t_n in seconds for a particular run are plotted. The total times T_{20} up to $n = 20$ for the two methods are also compared. Timings vary slightly from run to run.

the Clenshaw–Curtis method, that requires 55 integrand evaluations per interval, is replaced by the more expensive double-exponential method that uses up to 266 integrand evaluations per interval and several intervals per iteration for the present example.

Acknowledgements

The author thanks the *Deutsche Forschungsgemeinschaft* and the *Fonds der Chemischen Industrie* for financial support.

References

[1] B.W. Char, K.O. Geddes, G.H. Gonnet, B.L. Leong, M.B. Monagan, S.M. Watt, Maple V Language Reference Manual, Springer, Berlin, New York, 1991.
[2] B.W. Char, K.O. Geddes, G.H. Gonnet, B.L. Leong, M.B. Monagan, S.M. Watt, Maple V Library Reference Manual, Springer, Berlin, New York, 1991.