Applied Mathematics Letters 29 (2014) 20–25

Contents lists available at ScienceDirect

Applied Mathematics Letters

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

A simple modification of Newton's method to achieve convergence of order $1+\sqrt{2}$

Trevor J. McDougall [a,*], Simon J. Wotherspoon [b,c]

[a] School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2052, Australia

[b] Institute for Marine & Atmospheric Studies, Private Bag 129, University of Tasmania, Hobart, TAS 7001, Australia

[c] Australian Antarctic Division, 203 Channel Highway, Kingston, TAS 7050, Australia

ARTICLE INFO

Article history:
Received 22 August 2013
Received in revised form 16 October 2013
Accepted 16 October 2013

Keywords:
Newton's method
Non-linear equations
Root-finding
Iterative methods

ABSTRACT

A simple modification to the standard Newton method for approximating the root of a univariate function is described and analyzed. For the same number of function and derivative evaluations, the modified method converges faster, with the convergence order of the method being $1+\sqrt{2} \approx 2.4$ compared with 2 for the standard Newton method. Numerical examples demonstrate the faster convergence achieved with this modification of Newton's method. This modified Newton–Raphson method is relatively simple and is robust; it is more likely to converge to a solution than are either the higher order (4th order and 6th order) schemes or Newton's method itself.

1. Introduction

Newton's method for determining a root of a nonlinear equation $f(x) = 0$ has long been favored for its simplicity and fast rate of convergence. Using only the function and its first derivative, Newton's method iteratively produces a sequence of approximations that converge quadratically to a simple root. While a number of rules with higher order convergence have long been known [1], these have the disadvantage of requiring higher order derivatives.

Recently, a number of authors including [2–6] have derived multistep, predictor–corrector methods that offer higher order convergence but require only the function and its first derivative. These methods all require more function or derivative evaluations per iteration than Newton's method, but this additional cost is offset by the higher rate of convergence. Ref. [7] presents a similar predictor–corrector method that was subsequently employed in [8], and is now also used in several functions of the Gibbs Seawater (GSW) Oceanographic Toolbox of the International Thermodynamic Equation of Seawater (TEOS-10); see [9,10]. The method presented in [7] was motivated by an unusual application in which not only was there a good initial estimate of the root from which to start the iteration, but there was also a good estimate of the first derivative at that point. This extra derivative information, combined with the observation that if $f(x)$ is quadratic with root r then

$$r = x_0 - \frac{f(x_0)}{f'\left(\frac{1}{2}[x_0 + r]\right)}, \tag{1}$$

* Corresponding author. Tel.: +61 2 9385 3498; fax: +61 2 9385 7123; +61 407 518 183 (Mob.).
E-mail address: Trevor.McDougall@unsw.edu.au (T.J. McDougall).

0893-9659
http://dx.doi.org/10.1016/j.aml.2013.10.008

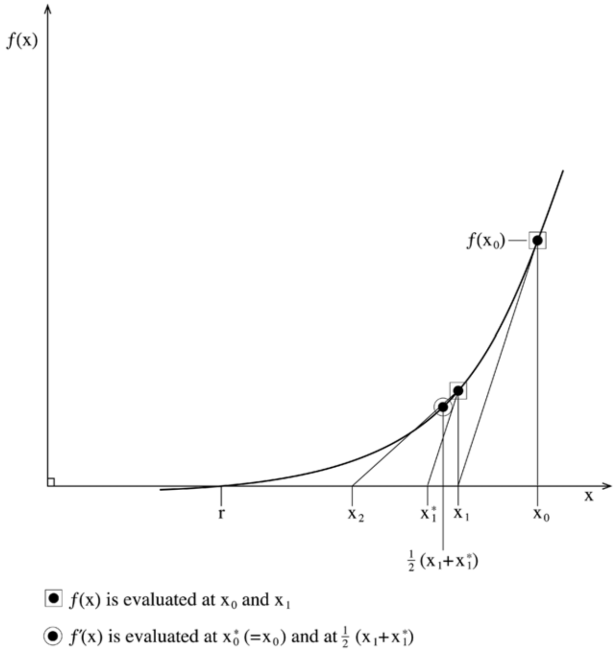

Fig. 1. Sketch of the modified Newton–Raphson method of this paper. The initial iteration to find x_1 is the standard Newton–Raphson scheme. But to find x_2 the function's derivative is not evaluated at x_1. Rather, the existing estimate of the derivative is re-used to find an intermediate value, x_1^*, and the derivative is evaluated at $\frac{1}{2}(x_1 + x_1^*)$. We call the step to find the intermediate value, x_1^*, the "predictor step", while the step to find the next value of x, x_2, (using the derivative evaluated at $\frac{1}{2}(x_1 + x_1^*)$) is called "the corrector" step.

suggests a predictor–corrector rule of the form

$$x_0^* = x_0 - f(x_0)/f_0' \tag{2a}$$

$$x_1 = x_0 - \frac{f(x_0)}{f'\left(\frac{1}{2}\left[x_0 + x_0^*\right]\right)}, \tag{2b}$$

where f_0' represents the estimate of the derivative at x_0. The initial predictor step is just a Newton step based on the estimated derivative, while the corrector step is motivated by the implicit relation (1). Subsequent iterations of the [7] method are similar, except that in the predictor step the estimated derivative is replaced by the derivative computed in the previous iteration, that is (starting at $k = 1$),

$$x_k^* = x_k - \frac{f(x_k)}{f'\left(\frac{1}{2}\left[x_{k-1} + x_{k-1}^*\right]\right)} \tag{3a}$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'\left(\frac{1}{2}\left[x_k + x_k^*\right]\right)}. \tag{3b}$$

It is this re-use in Eq. (3a) of the derivative calculated in the previous iteration that makes the rule particularly efficient — each full predictor–corrector step requires only one function and one derivative evaluation.

Here, the [7] method is extended to a more general context and it is shown that the rule converges at a rate of $1 + \sqrt{2}$ at a cost of one function and one derivative evaluation per full step, making this rule more efficient than the standard Newton–Raphson rule for the same cost.

2. The modified Newton–Raphson method

Once the iteration is established, our general iteration takes the form (3a) and (3b) above. As already noted, this is a predictor–corrector rule in which the predictor step is based on the derivative calculated in the previous iteration, and the corrector step is motivated by the implicit relation (1). The unusual feature of this scheme is the interleaving of function and derivative evaluations, that is, the function and its derivative are evaluated at different values of x (see Fig. 1).

Initializing the iteration essentially requires two starting values, x_0 and x_0^*. The iteration then proceeds with the corrector step (2b). Given an initial estimate x_0 of the root, two obvious methods for obtaining the second starting value x_0^* are to

- derive x_0^* by Newton's method $x_0^* = x_0 - f(x_0)/f'(x_0)$, or
- simply set $x_0^* = x_0$ in which case the corrector step (2b) for x_1 reduces to Newton's method.

In practice these two options are found to be equally effective, and henceforth only the $x_0^* = x_0$ case will be considered.

In summary, our general modification of Newton's method that we examine herein is given by

$$x_0^* = x_0 \tag{4a}$$

$$x_1 = x_0 - \frac{f(x_0)}{f'\left(\frac{1}{2}\left[x_0 + x_0^*\right]\right)} = x_0 - \frac{f(x_0)}{f'(x_0)}, \tag{4b}$$

followed by (for $k \geq 1$)

$$x_k^* = x_k - \frac{f(x_k)}{f'\left(\frac{1}{2}\left[x_{k-1} + x_{k-1}^*\right]\right)} \tag{5a}$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'\left(\frac{1}{2}\left[x_k + x_k^*\right]\right)}. \tag{5b}$$

These steps in our modified Newton–Raphson procedure are illustrated in Fig. 1, where the steps are shown up to finding x_2. The key features of our modified Newton–Raphson method can be seen in Fig. 1, namely

1. the value x_2 is calculated from x_1 using $f(x_1)$ and the value of the derivative evaluated at $\frac{1}{2}\left(x_1 + x_1^*\right)$ (which is a more appropriate value of the derivative to use than the one evaluated at x_1), and
2. this same value of the derivative is re-used in the next predictor step (not illustrated in Fig. 1) to obtain x_3^*.

This re-use of the derivative means that the evaluations of the starred values of x in Eq. (5a) essentially come for free, which then enables the more appropriate value of the derivative to be used in the corrector step Eq. (5b).

3. Convergence

Writing $x_i = r + \varepsilon_i$ and $x_i^* = r + \varepsilon_i^*$, repeated substitution and Taylor expansion in Eq. (5) shows that to leading order

$$\begin{array}{ll} & \varepsilon_0^* = \varepsilon_0 \\ \varepsilon_1 = \alpha\varepsilon_0^2 & \varepsilon_1^* = 2\alpha^2\varepsilon_0^3 \\ \varepsilon_2 = 2\alpha^4\varepsilon_0^5 & \varepsilon_2^* = 2\alpha^6\varepsilon_0^7 \\ \varepsilon_3 = 2^2\alpha^{11}\varepsilon_0^{12} & \varepsilon_3^* = 2^3\alpha^{16}\varepsilon_0^{17} \\ \varepsilon_4 = 2^5\alpha^{28}\varepsilon_0^{29} & \varepsilon_4^* = 2^7\alpha^{40}\varepsilon_0^{41} \\ \varepsilon_5 = 2^{12}\alpha^{69}\varepsilon_0^{70} & \ldots. \end{array} \tag{6}$$

where $\alpha = \frac{1}{2}f''(r)/f'(r)$. It is convenient to form the non-dimensional error measures (using over-tildes) as

$$\tilde{\varepsilon}_0 = \alpha\varepsilon_0, \qquad \tilde{\varepsilon}_i = \alpha\varepsilon_i \quad \text{and} \quad \tilde{\varepsilon}_i^* = \alpha\varepsilon_i^*, \tag{7}$$

so that Eqs. (6) become

$$\begin{array}{ll} & \tilde{\varepsilon}_0^* = \tilde{\varepsilon}_0 \\ \tilde{\varepsilon}_1 = \tilde{\varepsilon}_0^2 & \tilde{\varepsilon}_1^* = 2\tilde{\varepsilon}_0^3 \\ \tilde{\varepsilon}_2 = 2\tilde{\varepsilon}_0^5 & \tilde{\varepsilon}_2^* = 2\tilde{\varepsilon}_0^7 \\ \tilde{\varepsilon}_3 = 2^2\tilde{\varepsilon}_0^{12} & \tilde{\varepsilon}_3^* = 2^3\tilde{\varepsilon}_0^{17} \\ \tilde{\varepsilon}_4 = 2^5\tilde{\varepsilon}_0^{29} & \varepsilon_4^* = 2^7\tilde{\varepsilon}_0^{41} \\ \tilde{\varepsilon}_5 = 2^{12}\tilde{\varepsilon}_0^{70} & \ldots. \end{array} \tag{8}$$

Examining the sequence of powers over pairs of successive steps shows the approximation converges faster than quadratic. This is most easily seen by noting that the powers of $\tilde{\varepsilon}_0$ (and of 2) in these expressions for $\tilde{\varepsilon}_i$ progress in the series

$$2,\ 5,\ 12,\ 29,\ 70,\ 169,\ 408,\ 985,\ 2378,\ \ldots \tag{9}$$

and the successive ratios of these powers are

$$2.5,\ 2.4,\ 2.4167,\ 2.4138,\ 2.4143,\ 2.4142,\ 2.4142,\ 2.4142,\ \ldots. \tag{10}$$

The ratio R of consecutive numbers in this sequence is seen to rapidly approach a fixed number of approximately 2.4142. The Taylor series calculations above show that the numbers N_i in the sequence (9) are related by

$$N_i = 2N_{i-1} + N_{i-2}, \tag{11}$$

and dividing (11) by N_{i-1} and setting (in the limit) $N_i/N_{i-1} = N_{i-1}/N_{i-2} = R$, we see that R is the solution of $R = 2 + R^{-1}$ namely $R = 1 + \sqrt{2} \approx 2.4142$. The same asymptotic ratio R of consecutive terms is found from the sequence of powers

$$1,\ 3,\ 7,\ 17,\ 41,\ 99,\ 239,\ 577,\ 1393,\ \ldots \tag{12}$$

that appears in the right-hand side of Eq. (8), namely the "starred" expressions of (8).

Hence we conclude that our modified Newton–Raphson method of solving $f(x) = 0$ converges at the power $R = 1 + \sqrt{2} \approx 2.4142$. That is, $\tilde{\varepsilon}_i$ converges to $\tilde{\varepsilon}_{i-1}$ raised to the power $1 + \sqrt{2}$, and $\tilde{\varepsilon}_i^*$ also converges to $\tilde{\varepsilon}_{i-1}^*$ raised to the power $1 + \sqrt{2}$. By comparison, the standard Newton method converges quadratically, that is, at the power 2. For example, after

Table 1
Numerical efficiencies of various numerical schemes.

Method	Number of function or derivative evaluations	Efficiency index — convergence order per function evaluation
Newton, quadratic	2	$2^{1/2} \approx 1.4142$
Cubic methods	3	$3^{1/3} \approx 1.4422$
Kou's 5th order	4	$5^{1/4} \approx 1.4953$
The present paper	2	$\left(\sqrt{2}+1\right)^{1/2} \approx 1.5538$
Kou's 6th order	4	$6^{1/4} \approx 1.5651$
Jarratt's 4th order	3	$4^{1/3} \approx 1.5874$
Secant	1	$0.5\left(1+\sqrt{5}\right) \approx 1.6180$

four full iterations of the standard Newton–Raphson method, the error is $\tilde{\varepsilon}_0^{16}$ whereas the error remaining in our modified method is $\tilde{\varepsilon}_4 = 2^5\tilde{\varepsilon}_0^{29}$. Both cases require four function evaluations and four derivative evaluations at this stage.

4. Comparison with "cubic", "4th order", "6th order" and "secant" methods

Refs. [2,5,3,11] have presented numerical schemes having cubic convergence. In each iteration of these numerical schemes three evaluations are required of either the function or its derivative. The best way of comparing these numerical schemes is to express the rate of convergence per function or derivative evaluation, the so-called "efficiency" of the numerical scheme. On this basis, the standard Newton–Raphson scheme has an efficiency of $2^{1/2} \approx 1.4142$, the modified Newton–Raphson method of the present paper has an efficiency of $\left(\sqrt{2}+1\right)^{1/2} \approx 1.5538$, while the cubic convergence methods have an efficiency of $3^{1/3} \approx 1.4422$.

These efficiencies can be compared with the secant method which has an efficiency of $0.5\left(1+\sqrt{5}\right) \approx 1.6180$. This is larger than the efficiency of our modified Newton–Raphson method, however the secant method has the drawback that, very close to the root, least-significant-bit (LSB) noise will affect the denominator and the stopping criterion will not always be able to avoid this unwanted behavior.

Kou [4] has developed several methods that each require two function evaluations and two derivative evaluations and these methods achieve an order of convergence of either five or six, so having efficiencies of $5^{1/4} \approx 1.4953$ and $6^{1/4} \approx 1.5651$ respectively. The larger of these two efficiencies is larger than that of the present modified Newton–Raphson scheme by 1%. However, in the next section we will show that these [4] methods converge to a solution less often than both the standard Newton–Raphson method and our modified Newton–Raphson method. In these [4] methods the denominator is a linear combination of derivatives evaluated at different values x, so that when the starting value of x is not close to the root, this denominator may go to zero and the methods may not converge. Of the four 6th order methods suggested in [4], if the ratio of the function's derivatives at the two values of x differ by a factor of more than three, then the method gives an infinite change in x. That is, the derivatives at the predictor and corrector stages can both be the same sign, but if their magnitudes differ by more than a factor of three, the method does not converge.

Jarratt [12] developed a 4th order scheme that requires only one function evaluation and two derivative evaluations, and similar 4th order schemes have been described by [6]. Jarratt's scheme is similar to those of [4] in that if the ratio of the derivatives at the predictor and corrector steps exceeds a factor of three, the method gives an infinite change in x.

The efficiencies of the schemes we have discussed are summarized in Table 1.

5. Numerical examples

Computational tests of the method were performed with the CLN multiprecision library [13], using 64-digit floating-point arithmetic. In Table 2 we compare the performance of four methods; Newton's method, the modified rules of [2,5], and our modified Newton–Raphson method using the four test functions

$$\begin{aligned}
f_1(x) &= \sin^2 x - x^2 + 1 \\
f_2(x) &= x^2 - e^x - 3x + 2 \\
f_3(x) &= xe^{x^2} - \sin^2 x + 3\cos x + 5 \\
f_4(x) &= e^{x^2+7x-30} - 1
\end{aligned} \tag{13}$$

also considered by [2,5]. The stopping criteria used for these comparisons are $|x_{n+1} - x_n| < \delta$ and $|f(x_{n+1})| < \delta$ where $\delta = 1.0\mathrm{e}{-27}$. In Table 2 it is seen that our modified Newton–Raphson procedure generally arrives at the accurately iterated solution with less function and derivative evaluations than the standard Newton–Raphson method and the cubic methods. Indeed the cubic methods are generally not superior to the standard Newton method, probably because of the need to do three function evaluations in each iteration, so that the cubic methods often need to complete such a full iteration whereas the standard Newton–Raphson method may reach the desired accuracy after its two function or derivative evaluations of one of its iterations.

Table 2
Numerical comparisons of the methods of Newton, [2] (labeled WF), [5] and the present paper. The columns represent the number of iterations N and the number of function or derivative evaluations N_f required to meet the stopping criteria, and the magnitude $|f(x)|$ of $f(x)$ at the final estimate x_n.

	Method	N	N_f	$\|f(x_{n+1})\|$	x_n
$f(x) = \sin^2 x - x^2 + 1$					
$x_0 = 1$					
	Newton	8	16	3.4e−101	
	Wang	5	15	2.0e−93	1.404491648215341226035086817786868077 18
	WF	5	15	8.9e−89	
	Present	7	14	8.8e−113	
$x_0 = 3$					
	Newton	8	16	2.0e−88	
	Wang	6	18	1.0e−202	1.404491648215341226035086817786868077 18
	WF	5	15	7.9e−181	
	Present	7	14	1.2e−129	
$f(x) = x^2 - e^x - 3x + 2$					
$x_0 = 2$					
	Newton	6	12	2.9e−55	
	Wang	5	15	5.1e−158	0.257530285439860760455367304937241781385
	WF	5	15	5.9e−103	
	Present	6	12	3.5e−107	
$x_0 = 3$					
	Newton	8	16	4.1e−104	
	Wang	5	15	3.1e−102	0.257530285439860760455367304937241781385
	WF	6	18	5.0e−151	
	Present	7	14	7.4e−122	
$f(x) = xe^{x^2} - \sin^2 x + 3\cos x + 5$					
$x_0 = -2$					
	Newton	10	20	3.8e−81	
	Wang	7	21	1.1e−170	−1.207647827130918927009416758356084097 76
	WF	7	21	2.0e−129	
	Present	9	18	3.6e−155	
$f(x) = e^{x^2+7x-30} - 1$					
$x_0 = 3.25$					
	Newton	10	20	5.5e−66	
	Wang	7	21	2.0e−138	3.0
	WF	7	21	1.7e−107	
	Present	9	18	1.2e−124	
$x_0 = 3.5$					
	Newton	14	28	1.2e−94	
	Wang	9	27	3.2e−110	3.0
	WF	10	30	–	
	Present	12	24	7.0e−136	

We performed a more extensive series of tests than is shown in Table 2, using all the nine test functions and their several starting values of x described in [2], making a total of 18 test cases. Here we report on the relative performance of the method of this paper compared with that of the other methods discussed above. In each of these 18 test cases, our method outperformed the standard Newton method, and in two of the cases, the standard Newton method did not converge to a solution within 50 iterations (100 function or derivative evaluations). Our method also outperformed the method of [3] in every case, it outperformed the [5] method in 16 out of the 18 cases, and it outperformed the [2] method in 14 of the 18 cases. Our measure of performance here is to note the number of function or derivative evaluations needed to reduce the error of the initial guess by 30 orders of magnitude.

The evaluation of the derivative in our method at the mid-point between the most recent solution and a predictor value of the solution (that is not unlike Newton's method) is the reason why our method converges to a solution more often than does the standard Newton method.

We also implemented the four 6th order schemes of [4] using these 18 cases, and found that in 8 of the 72 (= 4x18) cases these did not converge to a solution even after 50 iterations (involving 200 evaluations of either the function or its derivative). When these 6th order schemes converged they were on average superior to the present modified Newton method, with the method of the present paper outperforming these Kuo 6th order schemes in only 24 of the 72 cases.

Similarly, we implemented two of the 4th order Jarratt-type schemes, specifically, Eqs. (17) and (19) of [6]. Our modified Newton method outperformed these schemes in slightly more than half of the 36 cases (10 cases out of 18 for Eq. (17) and 9 cases out of 18 for Eq. (19) of [6]. Of more concern for these Jarratt-type schemes, 10 of the 36 cases of Eqs. (17) and (19) of [6] did not converge to a solution after 50 iterations (involving 150 evaluations of either the function or its derivative).

In summary, our method outperformed Newton's method and all of the cubic methods. The relatively robust nature of our modified Newton method is illustrated by the fact that in 2 of the 18 cases, Newton's method did not converge. Taking the higher order schemes of [4,6] as a whole, the method of this paper outperformed these higher order methods in 42 of the 108 cases, and in all cases it converged to the solution. These higher order schemes did not converge to a solution in a total of 18 of the 108 examples.

Another advantage of our method is that the stopping criteria can be implemented half way through a complete iteration, that is, after the starred stage. If the stopping criteria are satisfied at this stage, this obviates the need to evaluate the derivative in the corrector step and so halves the cost compared with having to complete the full iteration before applying the stopping criteria. This points to an inherent advantage of a method that has fewer function evaluations per iteration because when the stopping criteria are very close to being met, fewer function evaluations are required to achieve the remaining accuracy required.

These comparisons with the standard Newton method have counted a function evaluation and a derivative evaluation as being equally costly in a computational sense. If however the function is such that the user can supply an algorithm which evaluates both the function and its derivative for not much more computational expense than the computation of either the function or its derivative, then the standard Newton method would be preferred since in the standard Newton method the function and its derivative are evaluated at the same value of x.

We note that at a double root this modified Newton–Raphson method behaves the same as the standard Newton method, that is, it exhibits linear convergence. We have not proven mathematically that the method can be extended to multiple dimensions, but in practice, we have found it to work in two dimensions.

6. Conclusion

The modified Newton–Raphson method presented in this paper offers an increased rate of convergence over Newton's rule with no additional cost. In practice the modified method is found to offer greater efficiency in terms of total function evaluations than other so-called cubic convergence methods. It is the re-use of previously computed values of the derivative that gives our modified method its numerical efficiency compared with the standard Newton–Raphson method.

Our modified Newton–Raphson method is relatively simple and is robust; it is more likely to converge to a solution than are either the higher order (4th order and 6th order) schemes or Newton's method itself. A stopping criteria can be applied to our modified Newton–Raphson method after each evaluation of either the function or its derivative. This feature is an additional advantage in terms of overall computational efficiency compared with methods that require several function or derivative evaluations in order to complete a full iteration.

The efficiency index (see Table 1) of our modified Newton method is 10% larger than that of Newton's method itself ($1.5538/1.4142 \approx 1.099$). The proportional increase in this efficiency index in adopting the Jarratt-type 4th order schemes is an additional 2% ($1.5874/1.5538 \approx 1.022$). This rather small gain in efficiency afforded by the higher order schemes must be balanced against the less robust nature of these higher order schemes, in that the starting values must be more carefully chosen to ensure convergence to the root.

References

[1] T.R. Scavo, J.B. Thoo, On the geometry of halley's method, Amer. Math. Monthly 102 (5) (1995) 417–426.
[2] S. Weerakoon, T.G.I. Fernando, A variant of Newton's method with accelerated third-order convergence, Appl. Math. Lett. 13 (2000) 87–93.
[3] H. Homeier, A modified newton method for root finding with cubic convergence, J. Comput. Appl. Math. 157 (1) (2003) 227–230.
[4] J. Kou, The improvements of modified Newton's method, Appl. Math. Comput. 189 (2007) 602–609.
[5] P. Wang, A third-order family of Newton-like iteration methods for solving nonlinear equations, J. Numer. Math. Stoch. 3 (2011) 13–19.
[6] F. Soleymani, S.K. Khattri, S.K. Vanani, Two new classes of optimal Jarratt-type fourth-order methods, Appl. Math. Lett. 25 (2012) 847–853.
[7] T.J. McDougall, D.R. Jackett, D.G. Wright, R. Feistel, Accurate and computationally efficient algorithms for potential temperature and density of seawater, J. Atoms. Ocean. Technol. 20 (2003) 730–741.
[8] D.R. Jackett, T.J. McDougall, R. Feistel, D.G. Wright, S.M. Griffies, Algorithms for density, potential temperature, conservative temperature and freezing temperature of seawater, J. Atmos. Ocean. Technol. 23 (2006) 1709–1728.
[9] T.J. McDougall, P.M. Barker, Getting started with TEOS-10 and the Gibbs Seawater (GSW) Oceanographic Toolbox, 2011, p. 28, SCOR/IAPSO WG127, ISBN 978-0-646-55621-5. Available from www.TEOS-10.org.
[10] IOC, SCOR, IAPSO, *The International Thermodynamic Equation of Seawater—2010: Calculation and Use of Thermodynamic Properties,*. Intergovernmental Oceanographic Commission, Manuals and Guides No. 56, UNESCO (English), 2010, p. 196 Available from http://www.TEOS-10.org.
[11] M. Frontini, E. Sormani, Some variant of Newton's method with third-order convergence, Appl. Math. Comput. 140 (2003) 419–426.
[12] P. Jarratt, Some efficient fourth order multipoint methods for solving equations, BIT 9 (1969) 119–124.
[13] B. Haible, CLN, a class library for numbers, 1996 Available from http://www.ginac.de/CLN/.