Sanjay Kumar Khattri

How to increase convergence order of the Newton method to $2 \times m$?

Applications of Mathematics, Vol. 59 (2014), No. 1, 15--24

Persistent URL: `http://dml.cz/dmlcz/143595`

HOW TO INCREASE CONVERGENCE ORDER OF THE NEWTON METHOD TO $2 \times m$?

Sanjay Kumar Khattri, Haugesund

(Received May 20, 2011)

Cordially dedicated to the Professor Joseph F. Traub

Abstract. We present a simple and effective scheme for forming iterative methods of various convergence orders. In this scheme, methods of various convergence orders, such as four, six, eight and ten, are formed through a modest modification of the classical Newton method. Since the scheme considered is a simple modification of the Newton method, it can be easily implemented in existing software packages, which is also suggested by the presented pseudocodes. Finally some problems are solved, to very high precision, through the proposed scheme. Numerical work suggests that the presented scheme requires less number of function evaluations for convergence and it may be suitable in high precision computing.

Keywords: iterative method, fourth order convergent method, eighth order convergent method, quadrature, Newton method, convergence, nonlinear equation, optimal choice

MSC 2010: 65H05, 65D99, 41A25

1. Introduction

The most common and probably the most used method for finding a simple root γ, i.e. $f(\gamma) = 0$, of a nonlinear scalar equation

$$f(x) = 0, \tag{1.1}$$

is the Newton method. The classical Newton method is given as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, 3, \ldots \tag{1.2}$$

It is well documented and well known that the Newton method converges quadratically (see [1], [3], [4], [6]–[9], [11]–[13], [17]–[24], [26], [27], [29]–[31] and references therein). There exist many modifications of the Newton method to improve the convergence order [1], [3], [4], [6]–[9], [11]–[24], [26], [27], [29]–[31]. Higher order modifications of the Newton method which are free of the second or higher derivatives have been actively researched. For example, third order convergent methods free of the second or higher derivatives are presented in [8], [9], [19], [23], fourth order convergent methods are developed in [1], [4], [6], [7], [17], [18], [20], [21], [27], [29], sixth order methods are developed in [5], [25], [28] and eighth order methods are presented in [10] and references therein.

There exist various modifications of the Newton method. The main drawback, of these powerful methods, from the implementation point of view is their independent nature. For example, if one has a software package which solves nonlinear equations by the well-known fourth order Jarrat method [13], then one may find it difficult to modify this package to implement sixth order methods [5], [25], [28] or the eighth order methods [10].

In this work, we develop a scheme that improves the order of convergence of the Newton method (1.2) from 2 to $2 \times m$. Here $m = 1, 2, 3, \ldots$ The choice $m = 1$ will result in the classical Newton method. Thus through our scheme, one may develop 4th order, 6th order, 8th order, ... convergent iterative methods. One of the beautiful facts of our scheme is that one needs modest modifications in the most used classical Newton iterative method (1.2) for achieving higher convergence rates. It may be very effective when one wants to modify an existing software package for achieving higher convergence order. Let us now develop our scheme.

2. The technique and convergence order of its various methods

Before presenting our technique, first we will develop iterative methods of various convergence orders. Consider the 4th order convergent iterative method

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \tag{2.1}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\Big[1 + \frac{f(y_n)}{f(x_n)}\Big(1 + 2\frac{f(y_n)}{f(x_n)}\Big)\Big]. \tag{2.2}$$

The error equation for the above method is given as

$$e_{n+1} = -\frac{1}{12}\frac{c_2(12c_3c_1 - 60c_2^2)}{c_1^3}e_n^4 + O(e_n^5). \tag{2.3}$$

Here, $c_k = f^k(\gamma)/k!$, $e_n = x_n - \gamma$, and γ is a simple root of $f(x)$. A proof of convergence of the above fourth order method is presented next.

P r o o f. Using the Taylor series of $f(x)$ around the solution γ and taking into account that $f(\gamma) = 0$, we get

$$f(x_n) = \sum_{k=1}^{\infty} c_k e_n^k. \tag{2.4}$$

Furthermore, from the equation (2.4) we have

$$f'(x_n) = \sum_{k=1}^{\infty} k c_k e_n^{k-1}, \tag{2.5}$$

and through a simple calculation we arrive at

$$\frac{f(x_n)}{f'(x_n)} = e_n - \frac{c_2}{c_1}e_n^2 - 2\frac{c_3c_1 - c_2^2}{c_1^2}e_n^3 - \frac{3c_4c_1^2 - 7c_2c_3c_1 + 4c_2^3}{c_1^3}e_n^4 + O(e_n^5). \tag{2.6}$$

Substituting (2.6) in (2.1) yields

$$y_n - \gamma = \frac{c_2}{c_1}e_n^2 + 2\frac{c_3c_1 - c_2^2}{c_1^2}e_n^3 + \frac{3c_4c_1^2 - 7c_2c_3c_1 + 4c_2^3}{c_1^3}e_n^4 + O(e_n^5). \tag{2.7}$$

Expanding $f(y_n)$ around the solution γ and using (2.7), we obtain

$$\begin{aligned} f(y_n) = c_2e_n^2 - \frac{1}{6}\frac{-12c_3c_1 + 12c_2^2}{c_1}e_n^3 \\ + \frac{1}{24}\frac{72c_4c_1^2 - 168c_2c_3c_1 + 120c_2^3}{c_1^2}e_n^4 + O(e_n^5). \end{aligned} \tag{2.8}$$

From equations (2.4) and (2.8) we get

$$\frac{f(y_n)}{f(x_n)} = \frac{c_2}{c_1}e_n + \frac{2c_3c_1 - 3c_2^2}{c_1^2}e_n^2 - \frac{-3c_4c_1^2 + 10c_2c_3c_1 - 8c_2^3}{c_1^3}e_n^3 + O(e_n^4). \tag{2.9}$$

Now from equations (2.6), (2.9), and (2.2) we find that

$$e_{n+1} = -\frac{c_2(-5c_2^2 + c_3c_1)}{c_1^3}e_n^4 + O(e_n^5). \tag{2.10}$$

This proves that the method (2.2) converges quartically. □

Let us now consider the three step and sixth order convergent iterative method

$$
(2.11)\quad \begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big)\Big], \\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big) + \dfrac{f(z_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big)\Big]. \end{cases}
$$

The error equation for the above sixth order method is given as

$$
e_{n+1} = \frac{c_2(-11c_3c_1c_2^2 + 30c_2^4 + c_3^2c_1^2)}{c_1^5}e_n^6 + O(e_n^7).
$$

The convergence order of the above method can be easily established through the Maple software package. We notice that the method (2.11) requires evaluations of only three functions and one derivative during each iterative step. Let us now further consider the eighth order convergent iterative method

$$
(2.12)\quad \begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big)\Big], \\ p_n = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big) + \dfrac{f(z_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big)\Big], \\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big) + \dfrac{f(z_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big) \\ \qquad + \dfrac{f(p_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big)\Big]. \end{cases}
$$

The asymptotic error equation for the above eight order method is given as

$$
e_{n+1} = \frac{c_2(180c_2^6 - 96c_3c_1c_2^4 + 17c_3^2c_1^2c_2^2 - c_3^3c_1^3)}{c_1^7}e_n^8 + O(e_n^9).
$$

We notice that the eighth order method (2.12) requires evaluations of only four functions and one derivative during each iterative step. Based upon the similarity in methods (2.2), (2.11), and (2.12), let us consider the method

$$(2.13)\quad \begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big)\Big], \\ p_n = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big) + \dfrac{f(z_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big)\Big], \\ q_n = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big) + \dfrac{f(z_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big) \\ \qquad + \dfrac{f(p_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big)\Big], \\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big) + \dfrac{f(z_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big) \\ \qquad + \dfrac{f(p_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big) + \dfrac{f(q_n)}{f(x_n)}\Big(1 + 2\dfrac{f(y_n)}{f(x_n)}\Big)\Big]. \end{cases}$$

Through the Maple we verified that the above method is 10th order convergent, and the error equation for it is given as

$$e_{n+1} = \frac{c_2}{c_1^9}(1080c_2^8 - 756c_3c_1c_2^6 + 198c_3^2c_1^2c_2^4 - 23c_3^3c_1^3c_2^2 + c_3^4c_1^4)e_n^{10} + O(e_n^{11}).$$

We notice that the above tenth order method (2.13) requires evaluations of only five functions and one derivative per iterative step. Based upon the methods (2.2), (2.11), (2.12), and (2.13), we conjecture the existence of the following scheme for generating the iterative method of order $2 \times m$:

$$(2.14)\quad \begin{cases} y_1 = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ y_2 = x - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_1)}{f(x_n)}\Big(1 + 2\dfrac{f(y_1)}{f(x_n)}\Big)\Big], \\ y_3 = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_1)}{f(x_n)}\Big(1 + 2\dfrac{f(y_1)}{f(x_n)}\Big) + \dfrac{f(y_2)}{f(x_n)}\Big(1 + 2\dfrac{f(y_1)}{f(x_n)}\Big)\Big], \\ y_4 = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_1)}{f(x_n)}\Big(1 + 2\dfrac{f(y_1)}{f(x_n)}\Big) + \dfrac{f(y_2)}{f(x_n)}\Big(1 + 2\dfrac{f(y_1)}{f(x_n)}\Big) \\ \qquad + \dfrac{f(y_3)}{f(x)}\Big(1 + 2\dfrac{f(y_1)}{f(x_n)}\Big)\Big], \\ \qquad \vdots \end{cases}$$

$$
\begin{cases}
y_{m-1} = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_1)}{f(x_n)}\Big(1 + 2\dfrac{f(y_1)}{f(x_n)}\Big) \\
\qquad\qquad + \ldots + \dfrac{f(y_{m-2})}{f(x_n)}\Big(1 + 2\dfrac{f(y_1)}{f(x_n)}\Big)\Big], \\
x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}\Big[1 + \dfrac{f(y_1)}{f(x_n)}\Big(1 + 2\dfrac{f(y_1)}{f(x_n)}\Big) \\
\qquad\qquad + \ldots + \dfrac{f(y_{m-1})}{f(x_n)}\Big(1 + 2\dfrac{f(y_1)}{f(x_n)}\Big)\Big].
\end{cases}
$$

It may be noticed that a $2 \times m$ order method, formed by the above scheme, will require m functions $f(x_n)$ and one derivative $f'(x_n)$ evaluation during each iterative step. We have verified the above scheme through the Maple software package till $m = 10$. We see that for $m = 1$, the scheme produces the classical Newton method. Furthermore, we may notice that the above scheme is formed through a simple modification of the Newton method, and can be easily implemented in existing software packages for achieving higher convergence orders. Algorithm 1 presents a pseudocode for the Newton iterative method, while Algorithm 2 presents a pseudocode for the developed scheme.

Algorithm 1 Newton iterative method

while $|f(x_n)| < \varepsilon$ or $|x_{n+1} - x_n| < \varepsilon$ **do**
  $x_{n+1} = x_n - f(x_n)/f'(x_n)$
end while

Algorithm 2 New scheme with convergence order $2 \times m$

while $|f(x_n)| < \varepsilon$ or $|x_{n+1} - x_n| < \varepsilon$ **do**
  $x_{n+1} = x_n - f(x_n)/f'(x_n)$
  for $i = 1$ to $i < m$ step 1 **do**
    $x_{n+1} = x_{n+1} - f(x_{n+1})/f'(x_n)(1 + 2f(x_{n+1})/f(x_n))$
  end for
end while

Comparing Algorithms 1 and 2, we notice that the developed scheme can be easily incorporated into existing software packages through a simple loop.

3. Numerical work

The order of convergence ξ of an iterative method is defined as [2]

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^{\xi}} = c \neq 0.$$

Here, e_n is the error after n iterations of a method. Through a simple calculation, we may show that the approximate value of the computational order of convergence (COC) ϱ is given as [2]

$$\varrho \approx \frac{\ln|(x_{n+1}-\gamma)/(x_n-\gamma)|}{\ln|(x_n-\gamma)/(x_{n-1}-\gamma)|}.$$

All the computations reported here are done in the programming language C^{++}. For numerical precision, we are using ARPREC [32]. The ARPREC package supports arbitrarily high level of numerical precision [32]. In the program, the precision in decimal digits is set at 2005 with the command "mp::mp init(2005)" [32]. For convergence, it is required that the distance of two consecutive approximations $|x_{n+1}-x_n|$ be less than ε. And, the absolute value of the function $|f(x_n)|$, also referred to as residual, be less than ε. Apart from the convergence criteria, our algorithm also uses maximum allowed iterations as stopping criterion. Thus our algorithm stops if (i) $|x_{n+1}-x_n| < \varepsilon$, (ii) $|f(x_n)| < \varepsilon$, (iii) itr > maxitr. Here, $\varepsilon = 1\times 10^{-300}$, itr is the iteration counter for the algorithm and maxitr $= 100$. Algorithms 1 and 2 are tested for the following functions [10]:

$$\begin{aligned}
f_1(x) &= x^5 + x^4 + 4x^2 - 15, \quad \gamma \approx 1.347,\\
f_2(x) &= \sin(x) - x/3, \quad \gamma \approx 2.278,\\
f_3(x) &= 10x\mathrm{e}^{-x^2} - 1, \quad \gamma \approx 1.679,\\
f_4(x) &= \cos(x) - x, \quad \gamma \approx 0.739,\\
f_5(x) &= \mathrm{e}^{-x^2+x+2} - 1, \quad \gamma \approx -1.000,\\
f_6(x) &= \mathrm{e}^{-x} + \cos(x), \quad \gamma \approx 1.746,\\
f_7(x) &= \ln(x^2 + x + 2) - x + 1, \quad \gamma \approx 4.152,\\
f_8(x) &= \sin^{-1}(x^2 - 1) - x/2 + 1, \quad \gamma \approx 0.5948.
\end{aligned}$$

Here, γ is the approximate solution. We run Algorithm 2 for four values of m: $m = 1, 2, 3, 4$. Here, $m = 1$ corresponds to the classical Newton method. We choose the same initial guess as found in the article [10]. Thus, the reader may find it easier to compare performance of various methods presented in this work and reported in the article [10]. Table 1 reports the outcome of our numerical work. Table 1 reports (iterations required, number of function evaluations needed, COC during second last iteration) for the Newton method ($m = 1$), fourth order iterative method ($m = 2$), sixth order iterative method ($m = 3$) and eighth order iterative method ($m = 4$). Computational order of convergence reported in Table 1 was observed during the second last iteration.

$f(x)$	x_0	NM($m = 1$)	$m = 2$	$m = 3$	$m = 4$
$f_1(x)$	1.6	$(9, 18, 2)$	$(4, 12, 4)$	$(2, \mathbf{8}, 5.66)$	$(2, 10, 7.6)$
$f_2(x)$	2.0	$(23, 46, 2)$	$(10, 30, 4)$	$(7, \mathbf{21}, 5.9)$	$(6, 30, 7.9)$
$f_3(x)$	1.8	$(10, 20, 2)$	$(4, \mathbf{12}, 3.99)$	$(3, \mathbf{12}, 6.21)$	$(3, 15, 8.22)$
$f_4(x)$	1.0	$(9, 18, 2)$	$(4, 12, 3.99)$	$(3, 12, 5.90)$	$(2, \mathbf{10}, 8.10)$
$f_5(x)$	-0.5	$(11, 22, 2)$	$(5, \mathbf{15}, 3.99)$	$(4, 16, 5.99)$	$(3, \mathbf{15}, 6.75)$
$f_6(x)$	2.0	$(9, 18, 2)$	$(4, 12, 3.99)$	$(3, 12, 5.99)$	$(2, \mathbf{10}, 8.10)$
$f_7(x)$	3.2	$(10, 20, 2)$	$(4, \mathbf{12}, 3.99)$	$(3, \mathbf{12}, 6.19)$	$(3, 15, 8.19)$
$f_8(x)$	1.0	$(10, 20, 2)$	$(4, \mathbf{12}, 4.01)$	$(3, \mathbf{12}, 6.35)$	$(3, 15, 8.36)$

Table 1. (iterations, number of function evaluations, COC) for the Newton method ($m = 1$), fourth order method ($m = 2$), sixth order method ($m = 3$) and eight order iterative method ($m = 4$).

The following two important observations were made during numerical experimentations:

(1) In Table 1, the methods which require the least number of functional evaluations for convergence are marked in bold. We may see in Table 1 that for the five functions, out of eight functions, the choice $m = 3$ is optimal, while for the functions $f_4(x)$, $f_5(x)$, and $f_6(x)$ the choice $m = 4$ is optimal. We may also observe that the choice $m = 1$ (the Newton method) is not an optimal choice for any function.

(2) From Table 1, we notice that for the functions $f_3(x)$, $f_7(x)$, and $f_8(x)$ the sixth order ($m = 3$) and eighth ($m = 4$) order methods require the same number of iterative steps. Table 2 reports residual $|f(x_n)|$ during the last iterative step for all methods.

$f(x)$	$m = 1$	$m = 2$	$m = 3$	$m = 4$
$f_3(x)$	10^{-918}	10^{-637}	10^{-715}	10^{-1067}
$f_7(x)$	10^{-435}	10^{-872}	10^{-671}	10^{-1522}
$f_8(x)$	10^{-347}	10^{-744}	10^{-800}	10^{-1302}

Table 2. Residual ($|f(x_n)|$).

An iterative method of order r adds r correct significant digits to the approximation during each iteration. Therefore, higher order methods are very efficient in reducing the residual. In Table 2, we observe that the higher order methods are efficient in reducing the residual. And, these methods are preferred during high precision computation cf. [32].

4. Conclusions

In this work, we have developed a scheme for formulating higher order iterative methods. The scheme is based on a modest modification of the classical Newton method. The scheme can be easily incorporated in existing software packages for achieving higher order convergence rates as suggested by the presented pseudocodes. The developed scheme is also tested for finding zero of some functions. The presented numerical work shows that the most frequently used classical Newton method is not an optimal choice (at least not for the problems solved).

Acknowledgment. We are grateful to the referees for constructive remarks and suggestions which have enhanced our work.

References

[1] *I. K. Argyros, D. Chen, Q. Qian*: The Jarratt method in Banach space setting. J. Comput. Appl. Math. *51* (1994), 103–106.
[2] *C. Chun*: Construction of Newton-like iteration methods for solving nonlinear equations. Numer. Math. *104* (2006), 297–315.
[3] *C. Chun*: A geometric construction of iterative functions of order three to solve nonlinear equations. Comput. Math. Appl. *53* (2007), 972–976.
[4] *C. Chun*: Some fourth-order iterative methods for solving nonlinear equations. Appl. Math. Comput. *195* (2008), 454–459.
[5] *C. Chun, Y. Ham*: Some sixth-order variants of Ostrowski root-finding methods. Appl. Math. Comput. *193* (2007), 389-394.
[6] *C. Chun, Y. Ham*: Some fourth-order modifications of Newton's method. Appl. Math. Comput. *197* (2008), 654–658.
[7] *C. Chun, Y. Ham*: A one-parameter fourth-order family of iterative methods for nonlinear equations. Appl. Math. Comput. *189* (2007), 610–614.
[8] *M. Frontini, E. Sormani*: Some variant of Newton's method with third-order convergence. Appl. Math. Comput. *140* (2003), 419–426.
[9] *H. H. H. Homeier*: On Newton-type methods with cubic convergence. J. Comput. Appl. Math. *176* (2005), 425–432.
[10] *S. K. Khattri*: Optimal eighth order iterative methods. Math. Comput. Sci. *5* (2011), 237–243.
[11] *S. K. Khattri*: Newton-Krylov algorithm with adaptive error correction for the Poisson-Boltzmann equation. MATCH Commun. Math. Comput. Chem. *56* (2006), 197–208.
[12] *S. K. Khattri*: Altered Jacobian Newton iterative method for nonlinear elliptic problems. IAENG, Int. J. Appl. Math. *38* (2008), 108–112.

[13] *S. K. Khattri*: Two optimal families of iterative methods for solving nonlinear equations. Analysis, München *31* (2011), 305–312.
[14] *S. K. Khattri, I. K. Argyros*: Sixth order derivative free family of iterative methods. Appl. Math. Comput. *217* (2011), 5500–5507.
[15] *S. K. Khattri, T. Log*: Derivative free algorithm for solving nonlinear equations. Computing *92* (2011), 169–179.
[16] *S. K. Khattri, T. Log*: Constructing third-order derivative-free iterative methods. Int. J. Comput. Math. *88* (2011), 1509–1518.
[17] *S. K. Khattri, M. A. Noor, E. Al-Said*: Unifying fourth-order family of iterative methods. Appl. Math. Lett. *24* (2011), 1295–1300.
[18] *R. F. King*: A family of fourth-order methods for nonlinear equations. SIAM J. Numer. Anal. *10* (1973), 876–879.
[19] *J. Kou, Y. Li, X. Wang*: A modification of Newton method with third-order convergence. Appl. Math. Comput. *181* (2006), 1106–1111.
[20] *J. Kou, Y. Li, X. Wang*: Fourth-order iterative methods free from second derivative. Appl. Math. Comput. *184* (2007), 880–885.
[21] *J. Kou, Y. Li, X. Wang*: A composite fourth-order iterative method for solving nonlinear equations. Appl. Math. Comput. *184* (2007), 471–475.
[22] *A. M. Ostrowski*: Solution of Equations and Systems of Equation. Pure and Applied Mathematics 9. Academic Press, New York, 1960.
[23] *A. Y. Özban*: Some new variants of Newton's method. Appl. Math. Lett. *17* (2004), 677–682.
[24] *F.-A. Potra, V. Pták*: Nondiscrete Induction and Iterative Processes. Research Notes in Mathematics 103. Pitman Advanced Publishing Program, Boston, 1984.
[25] *H. Ren, Q. Wu, W. Bi*: New variants of Jarratt's method with sixth-order convergence. Numer. Algorithms *52* (2009), 585–603.
[26] *S. K. Sen, R. P. Agarwal, S. K. Khattri*: Computational pitfalls of high-order methods for nonlinear equations. J. Appl. Math. Inform. *30* (2012), 395–411.
[27] *J. R. Sharma, R. K. Goyal*: Fourth-order derivative-free methods for solving non-linear equations. Int. J. Comput. Math. *83* (2006), 101–106.
[28] *J. R. Sharma, R. K. Guha*: A family of modified Ostrowski methods with accelerated sixth order convergence. Appl. Math. Comput. *190* (2007), 111–115.
[29] *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.
[30] *J. F. Traub*: Iterative Methods for the Solution of Equations. 2nd ed. Chelsea Publishing Company, New York, 1982.
[31] *S. Weerakoon, T. G. I. Fernando*: A variant of Newton's method with accelerated third-order convergence. Appl. Math. Lett. *13* (2000), 87–93.
[32] ARPREC. C++/Fortran-90 arbitrary precision package. Available at http://crd.lbl.gov/~dhbailey/mpdist/.

Author's address: *Sanjay Kumar Khattri*, Stord/Haugesund University College, Haugesund, Norway, e-mail: `sanjay.khattri@hsh.no`.