International Mathematical Forum, 4, 2009, no. 6, 281 - 286

A Newton-Like Method for Convex Functions

Nenad Ujević [1], Nena Jović and Lucija Mijić

Department of Mathematics, University of Split
Teslina 12/III, 21000 Split, Croatia

Abstract

A Newton-like method for convex functions is derived. It is shown that this method can be better than the Newton method. Especially good results can be obtained if we combine these two methods. Illustrative numerical examples are given.

Mathematics Subject Classification: 65H05

Keywords: nonlinear equations, method, convex functions

1 Introduction

One of the most basic problems in Numerical Analysis (and of the oldest numerical approximation problems) is that of finding values of the variable x which satisfy $f(x) = 0$ for a given function f. The Newton method is the most popular method for solving such equations. Some historical points about this method can be found in [10].

In recent years a number of authors have considered methods for solving the nonlinear equations. For example, see [1]-[10].

In this paper we derive a Newton-like method for convex functions. It turns out that this new method can be better than the Newton method. In Section 2 we derive this method and give three algorithms: an algorithm for the new method, the well-known algorithm for the Newton method and an algorithm which combine these two methods. In Section 3 we give few numerical examples which show that the new method can be better than the Newton method. We specially emphasize that good results can be obtained if we combine these two methods.

[1] ujevic@pmfst.hr

2 The method and algorithms

We first describe the method for solving the single nonlinear equation $f(x) = 0$, where f is a convex function. For that purpose we need some preliminary results.

Let $f : [a, b] \to R$ be a convex function such that f' exists. Then we have

$$f(b) - f(a) \geq f'(a)(b - a) \tag{1}$$

and

$$f(b) - f(a) \leq f'(b)(b - a). \tag{2}$$

From the Taylor formula of the first order we have

$$f(x) = f(a) + f'(a)(x - a) + R_1(x)$$

and

$$f(x) = f(b) + f'(a)(x - b) + R_2(x),$$

where we consider the above formulas in the interval $[a, b]$. If a is close to b ($a \approx b$) then the remainder terms R_1 and R_2 are small and we can write

$$f(a) + f'(a)(x - a) \approx f(b) + f'(a)(x - b).$$

In fact, we now consider the equation

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

The solution is

$$x = \frac{f(b) - f(a) + af'(a) - bf'(b)}{f'(a) - f'(b)}. \tag{4}$$

We now wish to show that $a \leq x \leq b$. For that purpose we suppose that $f \in C^2(a, b)$ and $f''(x) \geq 0$, i.e. f is a convex function. Then the derivative f' is an increasing function such that $f'(a) - f'(b) < 0$. Thus the inequality

$$a \leq \frac{f(b) - f(a) + af'(a) - bf'(b)}{f'(a) - f'(b)} \tag{5}$$

is equivalent to the inequality

$$af'(a) - af'(b) \geq f(b) - f(a) + af'(a) - bf'(b)$$

and this last inequality is equal to (2). Hence, (5) holds.

Now we consider the inequality

$$\frac{f(b)-f(a)+af'(a)-bf'(b)}{f'(a)-f'(b)} \le b \tag{6}$$

which is equivalent to the inequality

$$f(b)-f(a)+af'(a)-bf'(b) \ge bf'(a)-bf'(b)$$

and this last inequality is equal to (1). Thus (6) holds.

In other words, we proved that

$$a \le x \le b. \tag{7}$$

It is not difficult to see that these inequalities become

$$a < x < b \tag{8}$$

if f is a strictly convex function.

We now consider the equation $f(x) = 0$ and suppose that there exists a unique zero point $x^* \in [a, b]$, $f(x^*) = 0$. If we calculate x by the formula (4) then we can consider this element as a first approximation of the zero point x^*. If $f(x)f(a) \le 0$ then we set $b_{new} = x$ and $a_{new} = a_{old} = a$. If $f(b)f(x) \le 0$ then we set $a_{new} = x$ and $b_{new} = b_{old} = b$. If f is a strictly convex function then (8) holds such that we have

$$b_{old} - a_{old} < b_{new} - a_{new}. \tag{9}$$

We repeat the above procedure until $b_{new} - a_{new} < \varepsilon$, where $\varepsilon > 0$ is a small given number. In such a way we get the sequence (x_k), where each $x_k(= x)$ is an end point of the interval $[a_{new}, b_{new}]$. Since (9) holds it is obvious that $\lim_{k\to\infty} x_k = x^*$.

Finally, we give a practical algorithm.

Algorithm 1 *(New method)*

(1) Choose $a, b \in R$ such that there exists a unique zero point $x^* \in [a, b]$. Choose also $\varepsilon > 0$, $n \in N$ and set $i = 0$.

(2) $i \to i + 1$

(3) Calculate

$$x = \frac{f(b)-f(a)+af'(a)-bf'(b)}{f'(a)-f'(b)}.$$

(4) If $x \notin [a, b]$ then set $x = (a + b)/2$ and go to (6).

(5) If $f(a)f(x) \le 0$ then set $b = x$.

If $f(x)f(b) \leq 0$ then set $a = x$.
(6) Calculate $e = b - a$.
(7) If $i > n$ then stop.
If $e < \varepsilon$ then stop.
(8) Go to (1).

The step (4) in the above algorithm is introduced because of a possible numerical instability.

We also give the well-known algorithm for the Newton method.

Algorithm 2 *(Newton method)*
(1) Choose $x \in R$, $\varepsilon > 0$, $n \in N$ and set $i = 0$.
(2) $i \to i + 1$
(3) Calculate $z = x - f(x)/f'(x)$.
(4) Calculate $e = |z - x|$.
(5) If $i > n$ then stop.
If $e < \varepsilon$ then stop.
(6) Set $x = z$ and go to (2).

Finally, we give a combined algorithm.

Algorithm 3 *(Combined method)*
(1) Apply Algorithm 1 with the step
(6a) If $|f(x)| < \eta$, where $\eta > 0$ is a given small number, then go to (2) else go to (7) of Algorithm 1.
(2) Apply Algorithm 2.

3 Numerical examples

In this section we give four numerical examples. We use the algorithms from the previous section with the following parameters: the tolerance of errors is $1.0E - 12$ and the parameter $\eta = 0.1$. If $[a, b]$ is an interval for Algorithm 1 then we choose $x_0 = b$ for Algorithm 2. All examples are taken from the references at the end of this paper.

Example 4 *Let $f(x) = \exp(1 - x) - 1$ and $a = -10$, $b = 7$. It is obvious that the exact solution of the equation $f(x) = 0$ is $x^* = 1$. (We choose the interval $[a, b]$ such that the end points are not close to the exact solution.) Algorithm 1 requires* 46 *iterations to find an approximate solution with the given tolerance; Algorithm 2 requires* 403 *iterations while Algorithm 3 requires* 20 *iterations.*

Example 5 *Let* $f(x) = x\exp(-x)$ *and* $a = -3$*,* $b = 2$*. It is obvious that the exact solution of the equation* $f(x) = 0$ *is* $x^* = 0$*. Algorithm 1 requires* 44 *iterations to find an approximate solution with the given tolerance; Algorithm 2 diverges while Algorithm 3 requires* 21 *iterations.*

Example 6 *Let* $f(x) = 1/x - 1$ *and* $a = 0.1$*,* $b = 4$*. It is obvious that the exact solution of the equation* $f(x) = 0$ *is* $x^* = 1$*. Algorithm 1 requires* 37 *iterations to find an approximate solution with the given tolerance; Algorithm 2 diverges while Algorithm 3 requires* 20 *iterations.*

Example 7 *Let* $f(x) = \exp(x^2 - 3x - 4) - 1$ *and* $a = -1.5$*,* $b = 0$*. It is obvious that one exact solution of the equation* $f(x) = 0$ *is* $x^* = -1$*. (We also see that in this example we choose the interval* $[a, b]$ *such that the end points are close to the exact solution.) Algorithm 1 requires* 34 *iterations to find an approximate solution with the given tolerance; Algorithm 2 requires* 377 *iterations while Algorithm 3 requires* 18 *iterations.*

References

[1] S. Abbasbandy, Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method, Appl. Math. Comput., 145, (2003), 887–893.

[2] S. Amat, S. Busquier, J. M. Gutierrez, Geometric constructions of iterative functions to solve nonlinear equations, J. Comput. Appl. Math., 157, (2003), 197–205.

[3] E. Babolian, J. Biazar, Solution of nonlinear equations by adomian decomposition method, Appl. Math. Comput., 132, (2002), 167–172.

[4] A. Ben-Israel, Newton's method with modified functions, Contemporary Math., 204, (1997), 39–50.

[5] F. Costabile, M. I. Gualtieri, S. Serra-Capizzano, An iterative method for the solutions of nonlinear equations, Calcolo, 30, (1999), 17–34.

[6] M. Frontini, Hermite interpolation and a new iterative method for the computation of the roots of non-linear equations, Calcolo, 40, (2003), 109–119.

[7] Mamta, V. Kanwar, V. K. Kukreja, Sukhjit Singh, On some third-order methods for solving nonlinear equations, Appl. Math. Comput., to appear.

[8] N. Ujević, A method for solving nonlinear equations, Appl. Math. Comput., (in press).

[9] S. Weerakoon, T. G. Fernando, A variant of Newton's method with accelerated third-order convergence, Appl. Math. Lett., 13, (2000), 87–93.

[10] T. Yamamoto, Historical development in convergence analysis for Newton's and Newton-like methods, J. Comput. Appl. Math., 124 (2000), 1-23.

Received: January, 2008