

A Newton-Monte-Carlo method for solving nonlinear scalar equations

BOGDAN GAVREA
(CLUJ-NAPOCA)

ABSTRACT. In this paper we present a simple modification of the Newton's method for solving nonlinear scalar equations. The family of methods obtained through this modification, maintains the second order of convergence. The method can be used in a Monte-Carlo type environment which results in convergence for cases where the “*standard*” Newton methods fail to converge. This paper originates in the recent work of Trevor J. McDougall and Simon J. Wotherspoon, [1].

KEY WORDS: Newton's method, Monte Carlo simulation, predictor-corrector methods

MSC 2010: 26D10, 46N30

1 Introduction

Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a function of class C^2 in a neighborhood of a root r that satisfies $f(r) = 0$ and $f'(r) \neq 0$. The goal of this note is to present a modified Newton-Raphson method for solving the nonlinear equation

[◇]Bogdan Gavrea, Technical University of Cluj-Napoca,
email: Bogdan.Gavrea@math.utcluj.ro

$$(1.1) \quad f(x) = 0.$$

The method has its origins in the recent work, [1], where a predictor-corrector modified Newton method of order $1 + \sqrt{2}$ was introduced. The authors from [1] started from the simple observation that if $f(x)$ is quadratic, then

$$(1.2) \quad r = x_0 - \frac{f(x_0)}{f'(\frac{1}{2}(x_0 + r))}.$$

The identity (1.2) was then used in designing both the predictor and corrector steps in the modified Newton's method of [1]. More precisely, the authors in [1] showed that the method defined by

$$\begin{aligned} x_0^* &= x_0, \\ x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)}, \\ x_k^* &= x_k - \frac{f(x_k)}{f'(\frac{1}{2}(x_{k-1} + x_{k-1}^*))} \text{ (predictor), } k \geq 1, \\ x_{k+1} &= x_k - \frac{f(x_k)}{f'(\frac{1}{2}(x_k + x_k^*))} \text{ (corrector), } k \geq 1 \end{aligned}$$

has order of convergence $1 + \sqrt{2}$ when x_0 is chosen sufficiently close to r . The method above uses in both the predictor and corrector steps one function and one derivative evaluation. If more derivative evaluations are used, then higher order methods can be designed as it was shown in [2] and [3] for example.

In this note we propose a family of modified Newton methods starting from the method proposed by McDougall and Wotherspon in [1]. The method described here uses a parameter $\gamma_k \in (0, 1)$, so that at each step k , $k \geq 1$, the derivative in the predictor and corrector step is evaluated at $(1 - \gamma_k)x_{k-1} + \gamma_k x_{k-1}^*$ and $(1 - \gamma_k)x_k + \gamma_k x_k^*$, respectively. More precisely

the iterations can be defined as follows,

$$(1.3) \quad x_0^* = x_0,$$

$$(1.4) \quad x_1 = x_0 - \frac{f(x_0)}{f'(x_0)},$$

$$(1.5) \quad x_k^* = x_k - \frac{f(x_k)}{f'((1 - \gamma_k)x_{k-1} + \gamma_k x_{k-1}^*)} \text{ (predictor), } k \geq 1,$$

$$(1.6) \quad x_{k+1} = x_k - \frac{f(x_k)}{f'((1 - \gamma_k)x_k + \gamma_k x_k^*)} \text{ (corrector), } k \geq 1$$

Remarks.

1. We note that if γ_k is set to $1/2$ for all values of k , then the method of McDougall and Wotherspoon is recovered.
2. One may generalize the method above even more by introducing different weighting factors for the predictor and corrector steps above. More precisely, a different parameter $\tilde{\gamma}_k$ may be used in the corrector step. For the sake of simplicity we, will use the same parameter γ_k for both predictor and corrector steps.
3. In what follows we plan to generate the parameters γ_k randomly. Therefore, γ_k will be considered a realization of a random variable Γ , that follows the uniform distribution on the interval $[0, 1]$, i.e., $\Gamma \sim Unif(0, 1)$. We note that the expectation of Γ equals $1/2$, i.e., $E[\Gamma] = \frac{1}{2}$.

2 Local convergence analysis

Let $f : \mathbb{R} \rightarrow \mathbb{R}$ and let r be a simple root of the nonlinear equation $f(x) = 0$, i.e.,

$$f(r) = 0 \text{ and } f'(r) \neq 0.$$

In what follows we assume that f is a C^2 function in a neighborhood of r . Let $\alpha \in \mathbb{R}$ be defined as follows

$$\alpha = \frac{f''(r)}{2f'(r)}.$$

We will denote by ϵ_k and ϵ_k^* the errors in x_k and x_k^* , respectively. More precisely,

$$\begin{aligned} x_k &= r + \epsilon_k \\ x_k^* &= r + \epsilon_k^* \end{aligned}$$

The error estimates for the first iterates can be easily computed to get

$$\begin{aligned} \epsilon_1 &= \alpha \epsilon_0^2 + \mathcal{O}(\epsilon_0^3) \\ \epsilon_1^* &= 2\alpha^2 \epsilon_0^3 + \mathcal{O}(\epsilon_0^4) \\ \epsilon_2 &= 2\alpha^3 \epsilon_0^4 \left(\frac{1}{2} - \gamma_1 \right) + 4\gamma_1 \alpha^4 \epsilon_0^5 + \mathcal{O}(\epsilon_0^6) \\ \epsilon_2^* &= 4\alpha^5 \left(\frac{1}{2} - \gamma_1 \right) (1 - \gamma_2) \epsilon_0^6 + \mathcal{O}(\epsilon_0^7). \end{aligned}$$

From the above equations we can see that when $\gamma_1 = \frac{1}{2}$ the asymptotic expansions of [1] are recovered for ϵ_2 and ϵ_2^* . Therefore for the case of $\gamma_k = \frac{1}{2}, \forall k$, the same analysis that was performed in [1] can be carried over and the method will converge locally with an order of convergence of $1 + \sqrt{2}$ as shown in [1]. In what follows we will assume that

$$\gamma_k \neq \frac{1}{2}, \forall k.$$

It is easy to see that in the asymptotic expansions the leading term of ϵ_k^* contains a bigger power of ϵ_0 , than ϵ_k , i.e., if $\epsilon_k = \mathcal{O}(\epsilon_0^p)$ then $\epsilon_k^* = \mathcal{O}(\epsilon_0^q)$ with $q > p$. We will use this in obtaining an asymptotic recurrence equation for ϵ_{k+1} and ϵ_k . From (1.6), we get

$$(2.1) \quad \epsilon_{k+1} = \epsilon_k - \frac{f'(x_k)}{f'((1 - \gamma_k)x_k + \gamma_k x_k^*)}.$$

For the denominator of the fraction appearing in (2.1), we use the following expansion

$$f'((1 - \gamma_k)x_k + \gamma_k x_k^*) = f'(r) + (1 - \gamma_k)f''(r)\epsilon_k + h.o.t.,$$

where *h.o.t.* stands for higher order terms. Therefore, we get for (2.1), the following asymptotic expansion

$$(2.2) \quad f'(r)\epsilon_{k+1} = f'(r)\epsilon_k + (1 - \gamma_k)f''(r)\epsilon_k^2 - f(x_k) + h.o.t.$$

Writing in (2.2),

$$\begin{aligned} f(x_k) &= f(r) + f'(r)\epsilon_k + \frac{f''(r)}{2}\epsilon_k^2 + h.o.t \\ &= f'(r)\epsilon_k + \frac{f''(r)}{2}\epsilon_k^2 + h.o.t, \end{aligned}$$

gives the following asymptotic recurrence relationship

$$(2.3) \quad \epsilon_{k+1}f'(r) - f''(r)\left(\frac{1}{2} - \gamma_k\right)\epsilon_k^2 + h.o.t. = 0,$$

which points to the asymptotic relationship

$$\epsilon_{k+1} \approx 2\alpha\left(\frac{1}{2} - \gamma_k\right)\epsilon_k^2,$$

showing that the order of (local) convergence is 2. We note that (2.3) uses the fact that $\gamma_k \neq \frac{1}{2}$ which is our working assumption for all values of k . When $\gamma_k = \frac{1}{2}, \forall k$, the local error analysis performed in [1] is recovered and the method will converge locally with order $1 + \sqrt{2}$.

3 A Newton-Monte-Carlo type algorithm

In what follows we present a simple algorithm that uses for γ_k uniform sampling in the interval $(0, 1)$. There are N runs each run starting from a given x_0 and from a deterministically computed x_1 .

Newton-Monte-Carlo Algorithm.

```

>User defined parameters:
  >N:= number of runs
  >maxIter:= maximum number of iterations in every run
  >tolFun:= tolerance used on the function values
>Let  $x_0^* = x_0$  and compute  $x_1$  from a pure Newton step
>for i=1:N
  >k=1;
  >repeat
    >randomly select  $\gamma_k$  from the uniform distribution on  $[0,1]$ ;
    >compute  $x_k^*$  according to (1.5);
    >compute  $x_{k+1}$  according to (1.6);
    > $k := k + 1$ ;
  >until  $k > \text{maxIter}$  or  $|f(x_k)| \leq \text{tolFun}$ 
>endfor

```

4 Numerical results

In what follows we present a few numerical experiments that were conducted and which are meant to answer the following questions:

- How well does the method perform when the test functions are “well behaved”?
- Is the method able to numerically compute solutions for functions with oscillatory behavior?

In the first part we present our results for, what we call, “well behaved functions”. These were also used as test functions in [1]. Three functions are tested using the Newton-Monte-Carlo method. The user defined constant

`tolFun` is set to `tolFun` = 10^{-9} , while the number of maximum iterations allowed is set to 150. All runs were succesfull, in the sense that the desired function tolerance was achieved in fewer iterations than the maximum allowed. For each test functions we have presented results on three runs in Table 1. The last column in this table gives the sample mean for the values of γ that were used in a single run. Just by analyzing the data we can see that even though the number of iterations is the same in all the runs presented, a (significantly) bigger value of $|\bar{\gamma} - \frac{1}{2}|$ results in a bigger residual value $|f(x_f)|$.

To test the performance of the method for functions with oscillatory behavior we have picked the function $f(x) = e^x \sin^2(5x) + 2x$, whose graph can be seen in Figure 1. The solution to $f(x) = 0$ is $x^* = 0$. The method of [1] fails to converge if it is started with $x_0 = 3$. We have performed a 5 run Newton-Monte-Carlo simulation ($N = 5$) starting from $x_0 = 3$. We have used a function tolerance `tolFun` = 10^{-9} . In Figure 1, we have plotted the graph of $f(x)$ as well as the iterates $(x_k, f(x_k))$, $k = 1, 2, \dots, 16$ generated by the proposed method in a selected successfull run. It can be noted that the iterates follow the same type of oscillatory behavior “jumping” from one region to another and eventually if the random selection is a “lucky” one, the iterates will get close to the solution. For the run shown in Figure 1, the last generated iterate x_{16} satisfies $f(x_{16}) = \mathcal{O}(10^{-16})$. We mention that for this run the expectation (arithmetic mean) of the sample $\{\gamma_1, \dots, \gamma_{16}\}$ is $\bar{\gamma} = 0.48545$.

5 Conclusions and further work

In this note we have presented a Newton-Monte-Carlo type method for solving nonlinear scalar equations. The method has its origins in the work [1] and has the advantage of obtaining an acceptable numerical solutions even for the case of functions exhibiting a highly oscillatory behavior. We have presented a brief local convergence analysis which shows that the method preserves the local second second order convergence of the metod.

Numerical results presented here address two questions: the case of “*well behaved functions*” as well as the case of highly oscillatory functions. The numerical experiments conducted show that the methods “closer” to the

method given in [1] are the ones that have the sample mean $\bar{\gamma}$ (significantly) closer to $\frac{1}{2}$. For these methods, the residual value at the final iterate is lower than for the ones having a (significantly) bigger deviation from $\gamma^* = \frac{1}{2}$.

$f(x), r, x_0$	Run ID	nr. iter.	$ f(x_f) $	$\bar{\gamma}$
$f(x) = e^{x^2+7x-30} - 1,$ $r = 3,$ $x_0 = 3.5$	1	8	2.80412e-10	0.61257
	2	8	6.22755e-11	0.51567
	3	8	2.08363e-10	0.60866
$f(x) = \sin^2(x) - x^2 + 1,$ $r \approx 1.40449,$ $x_0 = 3$	1	4	3.33066e-16	0.68311
	2	4	1.77635e-15	0.44461
	3	4	1.33066e-16	0.57242
$f(x) = x^2 - e^x - 3x + 2$ $r \approx 0.25753,$ $x_0 = 3$	1	4	6.66133e-16	0.56598
	2	4	1.34159e-12	0.37360
	3	4	1.99840e-15	0.60016

Table 1: Results for some nonlinear equations exploited in [1].

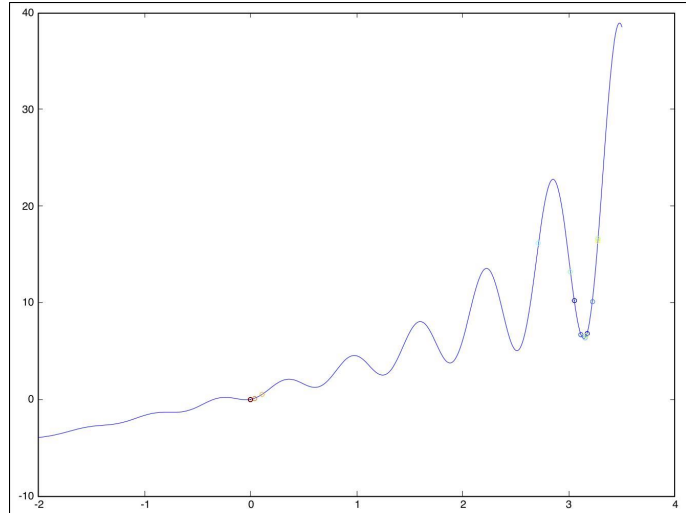


Figure 1: One Newton-Monte-Carlo run for solving the equation $e^x \sin^2(5x) + 2x = 0$, starting from $x_0 = 3$. The run terminates after 16 full iterates are performed with the function value $f(x_{16}) = O(10^{-16})$.

References

- [1] T. J. McDougall, S. J. Wothersponn, *A simple modification of Newton's method to achieve convergence of order $1 + \sqrt{2}$* , Applied Mathematics Letters, 29, pp. 20-25, 2014.
- [2] H.H.H. Homeier, *A modified Newton method for rootfinding with cubic convergence*, Journal of Computational and Applied Mathematics, 157, pp. 227-230, 2003.
- [3] J. Kou, *The improvements of modified Newtons method*, Applied Mathematics and Computation, 189(1), pp. 602-609, 2007.