

Article

Beyond Newton: A New Root-Finding Fixed-Point Iteration for Nonlinear Equations

Ankush Aggarwal ^{1,*}  and Sanjay Pant ² 

¹ Glasgow Computational Engineering Centre, James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, UK

² Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Swansea SA1 8EN, UK; sanjay.pant@swansea.ac.uk

* Corresponding; ankush.aggarwal@glasgow.ac.uk; Tel.: +44-141-330-4857

Received: 3 March 2020; Accepted: 27 March 2020; Published: 29 March 2020



Abstract: Finding roots of equations is at the heart of most computational science. A well-known and widely used iterative algorithm is Newton’s method. However, its convergence depends heavily on the initial guess, with poor choices often leading to slow convergence or even divergence. In this short note, we seek to enlarge the basin of attraction of the classical Newton’s method. The key idea is to develop a relatively simple multiplicative transform of the original equations, which leads to a reduction in nonlinearity, thereby alleviating the limitation of Newton’s method. Based on this idea, we derive a new class of iterative methods and rediscover Halley’s method as the limit case. We present the application of these methods to several mathematical functions (real, complex, and vector equations). Across all examples, our numerical experiments suggest that the new methods converge for a significantly wider range of initial guesses. For scalar equations, the increase in computational cost per iteration is minimal. For vector functions, more extensive analysis is needed to compare the increase in cost per iteration and the improvement in convergence of specific problems.

Keywords: Newton–Raphson method; Newton’s iteration; nonlinear equations; iterative solution; gradient-based methods

1. Introduction

Newton–Raphson, often referred to as Newton’s, fixed-point iteration method has been the gold standard for numerically solving equations for several centuries. To set the symbols and nomenclature, we start by defining a generic problem:

$$\text{Find } x = x^* \text{ such that } r(x) = 0 \quad (1)$$

for a given differentiable function $r : \mathbb{K} \rightarrow \mathbb{K}$ ($\mathbb{K} = \mathbb{R}$ or \mathbb{C}). The condition of differentiability is the same as that for Newton’s method. To solve this problem, fixed-point iterations are commonly used; starting with a guess to the solution $x_n = x_0$, the guess is iteratively updated using $x_{n+1} = \phi(x_n)$, where $\phi(x)$ depends on $r(x)$ and the chosen numerical scheme. It is required that $x_n \rightarrow x^*$ as $n \rightarrow \infty$ for the numerical scheme to converge to the true solution, and, in practice, the fastest possible convergence is desired. If we expand $r(x)$ in finite Taylor’s series up to second order about point x_n and evaluate it at $x = x^*$, we get

$$r(x^*) = r(x_n) + \left. \frac{dr}{dx} \right|_{x_n} (x^* - x_n) + \left. \frac{1}{2} \frac{d^2r}{dx^2} \right|_{\zeta} (x^* - x_n)^2 = 0, \quad (2)$$

for some $\zeta \in [x_n, x^*]$. The last term is called the remainder and can be written in other forms [1]. Since ζ is unknown, in the standard Newton’s method, we neglect the second order term and approximate x^* with an updated guess to the solution: x_{n+1} . Thus, we obtain Newton’s fixed-point iteration (using notation $r' = dr/dx$):

$$x_{n+1} = x_n + \Delta x^N = x_n - \frac{r(x_n)}{r'(x_n)}. \tag{3}$$

Using (2) and (3), one can rearrange to get the evolution of error $e_n = |x^* - x_n|$:

$$e_{n+1} = N(\zeta, x_n)e_n^2, \tag{4}$$

where $N(\zeta, x_n) := \left| \frac{r''(\zeta)}{2r'(x_n)} \right|$ is the measure of nonlinearity (This nonlinearity measure N is similar to the γ number in Smale’s α -theory[2]). The above step proves (at least) quadratic convergence of Newton’s method *close to the solution* when $r(x)$ has only simple roots. This quadratic convergence has led to a wide adoption of Newton’s method in solving problems in all scientific disciplines.

Despite the aforementioned attractive convergence of Newton’s method, it can be seen from (4) that to achieve quadratic convergence, the current guess x_n must be “sufficiently close” to the solution. When the initial guess is not close to the solution, the convergence can be slower than quadratic, or, worse, the iterations may diverge if $N(\zeta, x_n)e_n > 1$. As an example, if we consider $r(x) = e^x - 500$ and solve it using (3) with $x_0 = 0$, iterations diverge until we get numerical overflow (Figure 1a, solid red line). Instead, if we start with $x_0 = 3$, the error decreases sub-quadratically before achieving quadratic convergence closer to the solution (Figure 1a, dashed red line).

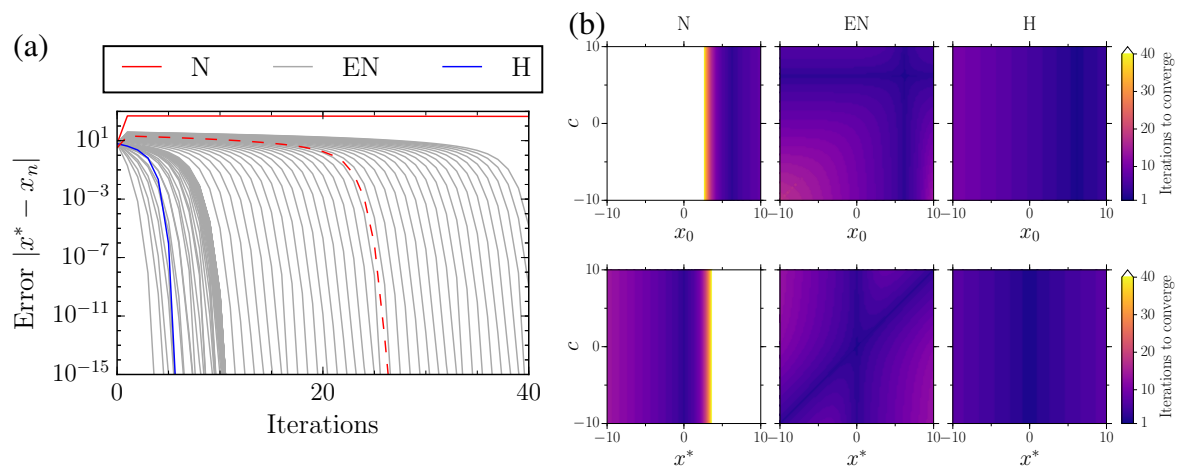


Figure 1. (a) Convergence of the standard Newton’s method (red lines), the Extended Newton method with $c \in (-50, 50)$ (gray lines), and the Halley’s method (blue line) for solving $e^x - 500 = 0$ (solid curves with initial guess $x_0 = 0$ and dashed curve with initial guess $x_0 = 3$). (b) (From left) Iterations taken to converge for the standard Newton’s method, the Extended Newton method, and the Halley’s method for solving $r(x) = e^x - H = 0$: with $H = 500$ and varying initial guess x_0 (top), and with $x_0 = 0$ and varying $H = e^{x^*}$ (bottom)

2. Methods and Results

The basin of attraction for a root x^* is defined as the set of initial guesses from which a given fixed-point iteration converges to x^* (this can be seen as the domain of convergence for initial guesses). Naturally, it is desirable to have a large basin of attraction for the root so that convergence is achieved for a wide range of initial guesses. For convergence we need $e_{n+1} < e_n$, i.e., $N(\zeta, x_n)e_n < 1$ using (4). Thus, size of the basin depends on nonlinearity $N(\zeta, x_n)$.

We pose the following question: can we pre-multiply the original equation by a function so that we obtain a larger (or, if possible, infinite) basin of attraction? That is, instead of solving (1), can we solve

$$\text{Find } x = x^* \text{ such that } P(x^*)r(x^*) = 0, \tag{5}$$

still using Newton's method? The idea is to choose a $P(x)$ that decreases the nonlinearity N and, hence, gives a larger basin of attraction, while retaining at least quadratic convergence close to the root. For (5), the measure of nonlinearity is $N(\zeta, x_n) = \left| \frac{[P(x)r(x)]''|_{x=\zeta}}{[P(x)r(x)]'|_{x=x_n}} \right|$. To minimize N , we equate $N = 0$, which gives

$$P(x) = \alpha \frac{x - c}{r(x)}, \tag{6}$$

for arbitrary integration constants α and c . We note that if $P(x)$ is not a function of x , we do not change the nonlinearity of the problem, and this case would fall under the purview of the highly developed field of linear preconditioning. Thus, without loss of generality, we set $\alpha = 1$. However, multiplying the above form of $P(x)$ by $r(x)$ has two problems: it (1) eliminates the original root $x = x^*$ and (2) introduces a new root $x = c$. To solve these problems, we add another constant $\kappa \neq 0$ in the denominator of $P(x)$ in (6), $P(x) = \frac{x - c}{r(x) + \kappa}$. This avoids the elimination of the x^* root. In addition, we eliminate the undesirable new root $x = c$ by choosing κ such that $r(c) + \kappa = 0$, i.e., $\kappa = -r(c)$ and $P(x) = \frac{x - c}{r(x) - r(c)}$. Thus, instead of applying Newton's method to $r(x) = 0$, if we apply Newton's method to our new equation

$$\frac{x - c}{r(x) - r(c)} r(x) = 0, \tag{7}$$

we get a new fixed-point iteration

$$x_{n+1} = x_n + \Delta x^{\text{EN}} = x_n - \frac{(x_n - c)r(x_n)}{r(x_n) - (x_n - c)r'(x_n) \frac{r(c)}{r(x_n) - r(c)}}. \tag{8}$$

The superscript ^{EN} denotes "Extended Newton" method.

2.1. Choice of c

For the $r(x) = e^x - 500$ example, the new equation to be solved is $(x - c)(e^x - 500)/(e^x - e^c) = 0$. Surprisingly, we find that our Extended Newton method, (8), converges starting from $x_0 = 0$ (Figure 1a, gray lines), irrespective of the choice of c in a wide range $c \in (-50, 50)$. Given the simplicity of the formulation, this is a remarkable result showing the potential of the EN method in comparison to the classical Newton's method.

The choice of c makes Extended Newton method a family of fixed-point iterations. We note that if $c = x^*$ such that $r(c) = 0$, we have $x_{n+1} = x_n + \Delta x^{\text{EN}} = x_n + (c - x_n) = c = x^*$, i.e., we will find the solution in a single iteration irrespective of the starting point x_0 and function r . This is not surprising; if one could choose $c = x^*$, the solution is already known, making the basin of attraction of EN method infinite.

We solve the problem $e^x - H = 0$ for varying values of c , x_0 , and H and find a significantly improved convergence using the proposed fixed-point iterations (Figure 1b). To verify the generality of the new scheme, we solve several different nonlinear equations and provide the results in the Supporting information (SI). We find that irrespective of the choice of c , the performance of EN is always better than the standard Newton's method. Therefore, we propose that the choice of c can be seen as another initial guess that the user needs to provide. We find that even if we choose c close to the starting guess x_0 , the convergence is greatly improved. Therefore, when no prior information on the choice of c is available, we recommend choosing $c = x_0 + \epsilon$ for small ϵ . This random perturbation

works well even in the case of multiple unknowns (see Section 2.3) and is proposed primarily due to convenience as provision of a guess for c is eliminated.

2.2. Limiting Case of $c \rightarrow x_n$

Next, we note that we must choose $c \neq x_n$. Choosing $c = x_n$ makes x_n a stationary point of the fixed-point iteration (8), and the iteration gets “stuck” at this undesired point. This is because, even though $\lim_{x \rightarrow c} P(x)r(x) \neq 0$, $P(c)r(c)$ is indeterminate. Practically, this does not pose a problem as long as $c \neq x_0$.

Nevertheless, we look at the limiting case when $c \rightarrow x_n$. Rewriting (8) and taking this limit we rediscover the classical Halley’s fixed-point iteration [3] (The same limit on our modified Equation (7) gives

$$\lim_{c \rightarrow x} \frac{(x - c)r(x)}{r(x) - r(c)} = \frac{r(x)}{r'(x)} = 0.$$

If we apply Newton’s method to the above equation, we get

$$\Delta x = -\frac{r(x_n)/r'(x_n)}{1 - r(x_n)r''(x_n)/r'(x_n)^2}$$

which is similar to Δx^H except for a factor of 2 in one term. This difference is likely due to the order in which the Newton iteration (which discards higher-order terms) and the limit are taken. Empirically, we have verified that (9) performs better than the above.)

$$\lim_{c \rightarrow x_n} \Delta x^{EN} = -\frac{r(x_n)/r'(x_n)}{1 - r(x_n)r''(x_n)/2r'(x_n)^2} \equiv \Delta x^H. \tag{9}$$

The advantage of this limit case is that there is no arbitrary constant involved; however, we have to pay with the price of calculating the second derivative of $r(x)$.

If we expand $r(x)$ using the Taylor’s series up to second order, using $\Delta x = x_{n+1} - x_n$, we get

$$r(x_n) + \left(r'(x_n) + \frac{r''(x_n)}{2} \Delta x \right) \Delta x = 0. \tag{10}$$

If we substitute Δx within the parentheses as the one from standard Newton (3) and rearrange, we get (9), i.e., (9) can be thought of as a correction to Newton’s equation using second derivative $r''(x)$. Thus, we can write:

$$r(x_n) + \left(r'(x_n) + \frac{r''(x_n)}{2} \Delta x^N \right) \Delta x^H = 0. \tag{11}$$

This property is useful in extending the Halley’s method to vector functions.

2.3. Extension to Multiple Unknowns

Extending to m unknowns x_i and m equations $r_i(x_1, \dots, x_m) = 0$ for $i = 1, \dots, m$, our problem is

$$\text{Find } x_i = x_i^* \text{ such that } r_i(\mathbf{x}^*) = 0 \forall i = 1, \dots, m, \tag{12}$$

where all m unknowns are collectively written as \mathbf{x} . Applying the standard Newton’s method gives us a system of m linear equations to be solved for step Δx_j^N

$$\sum_j r_{i,j}(\mathbf{x}) \Delta x_j^N = -r_i(\mathbf{x}), \tag{13}$$

where $r_{i,j} = \partial r_i / \partial x_j$ and we have dropped the subscript n to denote the current iteration value of \mathbf{x} as that is implied.

There could be several different ways of extending EN method (7) to vector functions. If it is possible to recognize a subset of the equations that are most nonlinear, one may use Extended Newton method for that subset of equations and standard Newton’s method for the rest. Here, we consider only the case where all equations are transformed. Thus, one straightforward extension is the following modified set of equations:

$$\frac{(x_i - c_i)r_i(\mathbf{x})}{r_i(\mathbf{x}) - r_i(\mathbf{x}_{c_i})} = 0 \tag{14}$$

for chosen values of c_i ($i = 1, \dots, m$), and \mathbf{x}_{c_i} is the vector \mathbf{x} with i -th element replaced by c_i . Applying Newton’s method we get our new system of linear equations

$$\sum_j \left\{ \delta_{ij} \frac{r_i(\mathbf{x})}{r_i(\mathbf{x}) - r_i(\mathbf{x}_{c_i})} + \frac{(x_i - c_i)}{[r_i(\mathbf{x}) - r_i(\mathbf{x}_{c_i})]^2} [r_{i,j}(\mathbf{x}_{c_i})r_i(\mathbf{x})(1 - \delta_{ij}) - r_{i,j}(\mathbf{x})r_i(\mathbf{x}_{c_i})] \right\} \Delta x_j^{\text{EN}} = -\frac{(x_i - c_i)r_i(\mathbf{x})}{r_i(\mathbf{x}) - r_i(\mathbf{x}_{c_i})}, \tag{15}$$

where $\delta_{ij} = 1$ when $i = j$ and 0 otherwise. For the single unknown case, we have to calculate $r(c)$ only once and there is no need for computing $r'(c)$. However, in the above extension to multiple unknowns, $r_i(\mathbf{x}_{c_i})$ needs to be evaluated at every iteration along with $r_{i,j}(\mathbf{x}_{c_i})$. Thus, the EN method for the case with multiple unknowns leads to a cumbersome set of calculations.

On the other hand, extension of the limit case of EN, the Halley’s method (11), to multiple unknowns is simpler:

$$\sum_j \left(r_{i,j} + \frac{1}{2} \sum_k r_{i,jk} \Delta x_k^{\text{N}} \right) \Delta x_j^{\text{H}} = -r_i, \tag{16}$$

where Δx_j^{N} is calculated from (13) and $r_{i,jk} = \frac{\partial^2 r_i}{\partial x_j \partial x_k}$. Thus, (13) and (16) can be solved successively.

To avoid solving two systems of m equations and instead obtain only one system of m equations, we multiply both sides of (16) by $r_{i,i}$. Using an approximation that $r_{i,i}r_{i,jk} \approx r_{i,k}r_{i,ji}$ and using (13), we get a simplified expression

$$\sum_j \left(r_{i,i}r_{i,j} - \frac{1}{2}r_{i,ji}r_i \right) \Delta x_j^{\text{QH}} = -r_i r_{i,i}. \tag{17}$$

This equation gives us the step Δx_j^{QH} by solving only one system of m equations, and we call it the quasi-Halley method (QH). In numerical methods, computing the Jacobian $r_{i,j}$ is common, but calculating another gradient $r_{i,jk}$ for (16) may not be desirable. However, in (17), we note that if we have an expression for $r_{i,j}$, $r_{i,ji}$ can be computed using finite difference in only $\mathcal{O}(m)$ operations and can be easily parallelized (unlike the iterations of Newton’s method).

As an extension of the exponential example from scalar function, we solve the following two equations:

$$[e^{x_1} - e^{x_2 - x_1}, e^{x_2 - x_1} - 1]^{\text{T}} = [0, 500]^{\text{T}}. \tag{18}$$

Similar to the single equation, the standard Newton’s method fails to converge for initial guess $x_1 = x_2 = 0$, whereas the Extended Newton (for $c_2 < -0.5$) and Halley’s methods converge in less than 10 iterations (Figure 2a).

As another example, we consider the Easom’s function of two variables

$$f(x, y) = -\cos(x) \cos(y) \exp(-x^2 - y^2) \tag{19}$$

and solve its derivative equal to zero:

$$\left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]^{\text{T}} = [0, 0]^{\text{T}}. \tag{20}$$

The solution of (20) lies at $(x, y) = (0, 0)$, and the basin of attraction is relatively small using the standard Newton’s method. Our proposed methods increase that basin significantly (Figure 2b). Surprisingly, the improvement using quasi-Halley’s method is even greater than the Halley’s method. Furthermore, Figure 2c demonstrates that choosing c as a perturbation to the initial guess x_0 also provides a significant improvement over the classical Newton’s method.

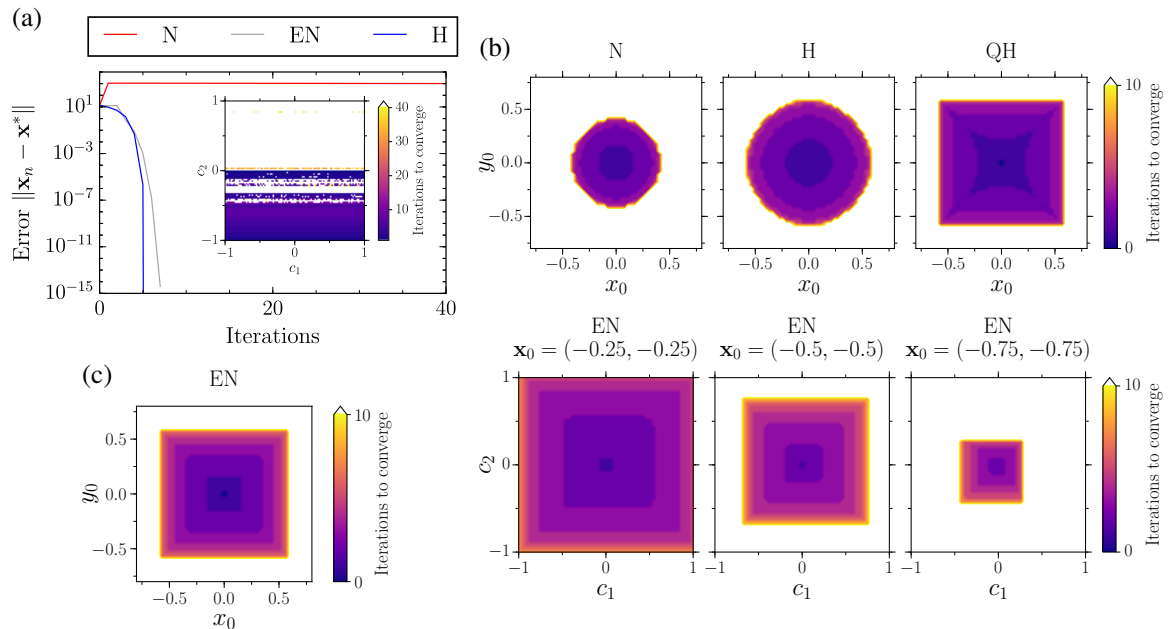


Figure 2. (a) Convergence for solving (18) with initial guess $x_1 = x_2 = 0$ using the standard Newton’s method (red line), the Extended Newton method with $c_1 = c_2 = -0.5$ (gray line), and the Halley’s method (blue line). The inset shows convergence of EN with the same initial condition and varying c_1 and c_2 . (b) Convergence in finding the minimum of the two-variable Easom’s function by solving its derivative equal to zero (20): (top) for the standard Newton’s, Halley’s and quasi-Halley methods with varying initial conditions, and (bottom) for Extended Newton method with varying c and three initial conditions. (c) Convergence in finding the minimum of the two-variable Easom’s function with varying initial guesses when c_i is a perturbation of initial guess, i.e., $c_1 = x_0 + \epsilon_1$ and $c_2 = y_0 + \epsilon_2$, where $\epsilon_{1,2} \in (-\frac{1}{2}10^{-4}, \frac{1}{2}10^{-4})$.

3. Discussion

A significant amount of work has focused on developing methods that approximate Newton’s method, largely through approximations of $r_{i,j}$ leading to the so-called quasi-Newton schemes with superlinear convergence [4]. On the other hand, super-quadratically convergent methods have been proposed using higher-order derivatives. For example, Halley’s method, Cauchy’s method, Chebyshev method [3], Householder’s method with arbitrary convergence [5], and other variants [6–9]. Newer higher-order methods are still being proposed [10], some of which are derivative-free [11]. However, the convergence of all these methods is examined when the starting guess is “sufficiently close” to the solution, also known as local convergence.

Different methods have been developed to analyze the semi-local convergence of iterative methods, such as by Kantorovich [12], Smale [2], and others [13–15]. Kantorovich method has been studied and extended by many [16–18]. However, these approaches only analyzed the semi-local convergence of existing methods, rather than developing new ones.

The objective of this article is to go beyond the classical approaches by decreasing the system nonlinearity in view of improving convergence properties *far from the solution*. Starting with a scalar equation, we pre-multiplied the original equation and formulated a new equation with the same roots (7). Applying Newton’s method to this equation, we proposed a new fixed-point iteration, which

we call the Extended Newton (EN) method (8). The proposed extensions are summarized in Table 1 and discussed next.

Table 1. Summary of Newton’s method and its proposed extensions

Method	Scalar Function Update	Vector Function Update
N	$\Delta x^N = -\frac{r(x_n)}{r'(x_n)}$	$\sum_j r_{i,j}(\mathbf{x}) \Delta x_j^N = -r_i(\mathbf{x})$
EN	$\Delta x^{EN} = -\frac{(x_n-c)r(x_n)}{r(x_n)-(x_n-c)r'(x_n)\frac{r(c)}{r(x_n)-r(c)}}$	$\sum_j \left\{ \delta_{ij} \frac{r_i(\mathbf{x})}{r_i(\mathbf{x})-r_i(\mathbf{x}_{ci})} + \frac{(x_i-c_i)}{[r_i(\mathbf{x})-r_i(\mathbf{x}_{ci})]^2} [r_{i,j}(\mathbf{x}_{ci})r_i(\mathbf{x})(1 - \delta_{ij}) - r_{i,j}(\mathbf{x})r_i(\mathbf{x}_{ci})] \right\} \Delta x_j^{EN} = -\frac{(x_i-c_i)r_i(\mathbf{x})}{r_i(\mathbf{x})-r_i(\mathbf{x}_{ci})}$
H	$\Delta x^H = -\frac{r(x_n)/r'(x_n)}{1-r(x_n)r''(x_n)/2r'(x_n)^2}$	$\sum_j \left(r_{i,j} + \frac{1}{2} \sum_k r_{i,jk} \Delta x_k^N \right) \Delta x_j^H = -r_i$
QH	–	$\sum_j \left(r_{i,i}r_{i,j} - \frac{1}{2}r_{i,ji}r_i \right) \Delta x_j^{QH} = -r_i r_{i,i}$

3.1. Extended-Newton Method

The application of EN method to a sample of scalar equations provided incredible improvements. The equation $e^x - H = 0$ could be solved for extremely high H in less than 10 iterations. Similar improvements were found for other examples (see SI). We also note that EN method provides at least quadratic convergence close to the root, because it is a Newton’s method applied to $P(x)r(x)$. For scalar functions, EN method does not increase the computational cost of each iteration and is a promising candidate. It should be noted that decreasing the nonlinearity can also be viewed as making the Jacobian Lipschitz, which is one of the parameters in the Kantorovich theorem [12]. The functions tested here are not natively Lipschitz in their domain; the proposed transformation makes them Lipschitz thus improving their semi-local convergence.

For a specific (known) equation form, instead of pre-multiplying, a transformation may be more suitable to decrease its nonlinearity. For example, for equation $e^x - H = 0$, one may write the equation as $e^x = H$ and then take a logarithm on both sides resulting in a linear equation that will always be solved in one iteration. However, this approach is problem-specific, requires identifying the two sides of the equation, and has been proposed by us for nonlinear elasticity [19]. On the other hand, the present approach is completely general and works for all kinds of equations and nonlinearities.

Although we used a constant value of c during iteration, there are other options of choosing its value at every iteration based on the current guess x_n , previous guess x_{n-1} , and/or the function form $r(x)$. In fact, if we choose $c = x_{n-1}$, the EN can be viewed as a combination of secant method and Newton’s method.

3.2. Rediscovery of Halley’s Method

Taking the limit $c \rightarrow x_n$, we rediscovered the Halley’s method (11), which is an extensively studied cubic-convergent method, yet rarely used in practice. Interestingly, although there are several cubic-convergent methods reported in the literature, only Halley’s method emerges naturally out of EN. One drawback of Halley’s method is that it needs the second derivative $r''(x)$. However, with classical approximation approaches such as the finite difference method, and the development of newer methods based on symbolic computation [20] and automatic differentiation [21], this limitation is becoming less important.

3.3. Vector Equations

Extension of EN method to vector functions can be accomplished in multiple ways. We presented one option (15) and found that the convergence improved significantly. However, the presented

approach entails additional computation (unlike for scalar functions). The extension of Halley's method to vector functions was straightforward and unique (16). Even though this approach worked perfectly, it has two computational disadvantages: calculation and storage of $r_{i,jk}$ with m^3 terms and solving two linear systems – first for the standard Newton (13) and then for the correction (16).

As a simplification, we proposed its approximation (17), the quasi-Halley method. This approach only requires calculation of one diagonal of $r_{i,jk}$, i.e., $r_{i,ji}$ and solving only one linear system of equations, making it an attractive option. To our knowledge, the quasi-Halley method has no hitherto reported counterpart. We believe that one of the reasons why these higher-order methods (Halley's or other higher-order methods) are not widely used in practice is that their faster convergence rates are offset by their higher computational cost per iteration. In this respect, the quasi-Halley method may provide a good balance between improved convergence properties and computational expense.

3.4. Future Directions

We believe there are many opportunities of extending this approach to other problems and end our discussion by mentioning a few potential avenues. For extending the EN method to boundary value problems, one could consider the premultiplication of the different (strong/weak/local conservation) forms and testing the convergence using different discretization methods. For extension to vector functions, approaches to design efficient $\mathbf{P}(\mathbf{x})$, which is a matrix, need to be developed. Similarly, this work motivates the need to explore quasi-Halley methods. In general, the extension of such root-finding methods to multiple-variables is a broad area, which also takes into account additional factors, such as sparsity and symmetry of the system, into account. The idea of multiplying with a function of x instead of a constant, can be used in the development of preconditioners. Unlike linear preconditioners, which are well-understood, this will lead to nonlinear preconditioning, which is a relatively under-explored area [22]. Application of a similar approach to nonlinear least squares problem will likely improve its convergence. Similarly, in optimization problems with multiple unknowns, a line search is commonly used, which is akin to solving a scalar equation. An approach similar to EN method may provide an improved line search.

3.5. Conclusions

It took several decades to develop methods based on the classical Newton's method. We hope that the new approach presented in this work, which resulted in some new and some known results, will spark the interest of the scientific community to fully explore the properties of these approaches, their applications, and the opportunities they present.

Supplementary Materials: The following are available online at <http://www.mdpi.com/1999-4893/13/4/78/s1>.

Author Contributions: Conceptualization, A.A. and S.P.; Formal analysis, A.A.; Investigation, A.A. and S.P.; Writing—original draft preparation, A.A.; Writing—review and editing, A.A. and S.P.; Funding acquisition, A.A. and S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Welsh Government and Higher Education Funding Council for Wales through the Sêr Cymru National Research Network in Advanced Engineering and Materials (Grant No. F28 to A.A.), and the Engineering and Physical Sciences Research Council of the UK (Grant No. EP/P018912/1 to A.A. and Grant No. EP/R010811/1 to S.P.).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Apostol, T.M. *Calculus*; Jon Wiley & Sons: New York, NY, USA, 1967.
2. Smale, S. Algorithms for solving equations. In Proceedings of the International Congress of Mathematicians, Berkeley, CA, USA, 3–11 August 1986; pp. 1263–1286.
3. Traub, J.F. *Iterative Methods for the Solution of Equations*; American Mathematical Soc.: Washington, WA, USA, 1982.
4. Kelley C.T. *Iterative Methods for Linear and Nonlinear Equations*; SIAM: Philadelphia, PA, USA, 1995.

5. Householder, A. *The Numerical Treatment of a Single Nonlinear Equation*; McGraw-Hill: New York, NY, USA, 1970.
6. Osada, N. An optimal multiple root-finding method of order three. *J. Comput. Appl. Math.* **1994**, *51*, 131–133.
7. Amat, S.; Busquier, S.; Gutiérrez, J. Geometric constructions of iterative functions to solve nonlinear equations. *J. Comput. Appl. Math.* **2003**, *157*, 197–205.
8. Kou, J. Some variants of Cauchy's method with accelerated fourth-order convergence. *J. Comput. Appl. Math.* **2008**, *213*, 71–78.
9. Cordero, A.; Torregrosa, J.R.; Vindel, P. Dynamics of a family of Chebyshev–Halley type methods. *Appl. Math. Comput.* **2013**, *219*, 8568–8583.
10. Sahu, D.R.; Agarwal, R.P.; Singh, V.K. A Third Order Newton-Like Method and Its Applications. *Mathematics* **2018**, *7*, 31.
11. Li, J.; Wang, X.; Madhu, K. Higher-Order Derivative-Free Iterative Methods for Solving Nonlinear Equations and Their Basins of Attraction. *Mathematics* **2019**, *7*, 1052.
12. Kantorovich, L. Functional analysis and applied mathematics. *Uspehi Mat. Nauk* **1948**, *3*, 89–185.
13. Dedieu, J.P.; Priouret, P.; Malajovich, G. Newton's method on Riemannian manifolds: Covariant alpha theory. *IMA J. Numer. Anal.* **2003**, *23*, 395–419.
14. Argyros, I.K.; Magreñán, Á.A.; Orcos, L.; Sarría, Í. Advances in the Semilocal Convergence of Newton's Method with Real-World Applications. *Mathematics* **2019**, *7*, 299.
15. Yong, Z.; Gupta, N.; Jaiswal, J.P.; Madhu, K. On the Semilocal Convergence of the Multi-Point Variant of Jarratt Method: Unbounded Third Derivative Case. *Mathematics* **2019**, *7*, 540.
16. Dennis, Jr, J. On the Kantorovich hypothesis for Newton's method. *SIAM J. Numer. Anal.* **1969**, *6*, 493–507.
17. Ortega, J.M. The newton-kantorovich theorem. *Am. Math. Mon.* **1968**, *75*, 658–660.
18. Tapia, R. The Kantorovich theorem for Newton's method. *Am. Math. Mon.* **1971**, *78*, 389–392.
19. Mei, Y.; Hurtado, D.E.; Pant, S.; Aggarwal, A. On improving the numerical convergence of highly nonlinear elasticity problems. *Comput. Methods Appl. Mech. Eng.* **2018**, *337*, 110–127.
20. Buchberger, B. Symbolic computation: Computer algebra and logic. In *Frontiers of Combining Systems*; Baader, F., Schulz, K.U., Eds.; Springer: Dordrecht, The Netherlands, 1996; Volume 3, pp. 193–219.
21. Rall, L. Automatic Differentiation: Techniques and Applications. *Lect. Notes. Comput. Sci.* **1981**, *120*, 173.
22. Cai, X.C.; Keyes, D.E. Nonlinearly preconditioned inexact Newton algorithms. *SIAM J. Sci. Comput.* **2002**, *24*, 183–200.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).