SIAM REVIEW
Vol. 36, No. 2, pp. 272–276, June 1994

007

ACCELERATED CONVERGENCE IN NEWTON'S METHOD*

JÜRGEN GERLACH†

Abstract. Newton's Method is based on a linear approximation of the function whose roots are to be determined taken at the current point, and the resulting algorithm is known to converge quadratically. In a procedure to increase the rate of convergence the author modifies the target function in such a way that Newton's Method applied to the modified function will yield a faster rate of convergence.

Key words. Newton's method, rate of convergence

AMS subject classifications. 65-01, 65B99, 65H05

This paper is concerned with achieving higher-order convergence for Newton's method. Newton's method itself converges quadratically, and it is based on a linear approximation (tangent line) to the function at the current iterate x_k:

$$f(x) \approx l(x) = f(x_k) + f'(x_k)(x - x_k) .$$

The equation $l(x_{k+1}) = 0$ leads to the familiar Newton scheme

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} . \tag{1}$$

In order to accelerate Newton's method, numerical analysis texts frequently suggest using a higher-order approximation to f at x_k; see, e.g., [2] or [3]. In this note we take a different approach, and ask: *For what class of functions does Newton's method perform particularly well?* And, secondly, once these functions have been identified we ask: *How can we modify a given function in such a way that the order of convergence is increased?*

Throughout we assume that

1. f is sufficiently many times differentiable;
2. f has a simple root at $x = a$, i.e., $f(a) = 0$ and $f'(a) \neq 0$; and that
3. the initial approximation x_0 is sufficiently close to a so that convergence to a will occur.

The answer to our first question is given by the following.

THEOREM 1. *In addition to the hypotheses above let us assume that* $f''(a) = f'''(a) = \cdots = f^{(m-1)}(a) = 0$, *and that* $f^{(m)}(a) \neq 0$. *Then the Newton sequence* $\{x_k\}$ *defined by* (1) *yields*

$$|x_{k+1} - a| \leq C|x_k - a|^m \tag{2}$$

for some constant C.

This theorem can be found as an exercise in the book by Dennis and Schnabel [2]; its proof is included in the Appendix. It roughly says that the more f looks like a linear function, the faster the Newton iterations will converge. Our next goal is to mold a given function into a new one in such a way that the roots remain unchanged, but it looks *nearly linear* in a neighborhood of the root so that the convergence of Newton's method will be accelerated.

Before we present a general result, let us begin with the case $f(a) = 0$, $f'(a) > 0$, and $f''(a) \neq 0$, and consider the function $F(x) := f(x)g(x)$, where the function $g(x)$ is to be

*Received by the editors January 14, 1993; accepted for publication August 10, 1993.

†Department of Mathematics and Statistics, Radford University, Box 6942, Radford, VA 24142.

determined subsequently. We have $F(a) = 0$ and, if $g(a) \neq 0$, we get $F'(a) \neq 0$. Moreover, since $f(a) = 0$,

$$F''(a) = 2f'(a)g'(a) + f''(a)g(a).$$

Although we only wish to satisfy the equation $F''(a) = 0$ at the single point $x = a$, we turn it into a differential equation for the function g:

$$g'(x) = -\frac{1}{2}\frac{f''(x)}{f'(x)}\, g(x).$$

Upon integration we obtain $g(x) = C/\sqrt{f'(x)}$ as a general solution, and with $C = 1$ the function F becomes $F(x) = f(x)/\sqrt{f'(x)}$. Newton's method applied to $F(x)$ yields

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)} = x_k - \frac{f(x_k)f'(x_k)}{f'(x_k)^2 - \frac{1}{2}f(x_k)f''(x_k)}. \tag{3}$$

This is an example of an iteration scheme with cubic convergence, and can be traced back to Halley in 1694 (see [4]).

Let us now turn to a general method, and present the result in the following form.

THEOREM 2. *Let* $f(a) = 0$, $f'(a) > 0$, $f''(a) = \cdots = f^{(m-1)}(a) = 0$, *and let* $f^{(m)}(a) \neq 0$. *Then the function*

$$F(x) = \frac{f(x)}{\sqrt[m]{f'(x)}} \tag{4}$$

satisfies $F(a) = 0$, $F'(a) > 0$, $F''(a) = \cdots = F^{(m)}(a) = 0$. *Moreover, for* $m \geq 3$ *we have* $F^{(m+1)}(a) = -\frac{1}{m}f^{(m+1)}(a)/\sqrt[m]{f'(a)}$.

Again we refer to the Appendix for the proof of this assertion. The determination of m may not always be easy. However, it is reassuring to know that the performance of Newton's method for $F(x) = f(x)/\sqrt[m]{f'(x)}$ will never be worse than the one found for $f(x)$ itself, regardless of the choice of m. The Newton iterations based on the function F are given by

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)} = x_k - \frac{f(x_k)f'(x_k)}{f'(x_k)^2 - \frac{1}{m}f(x_k)f''(x_k)}, \tag{5}$$

in analogy to (3).

Theorem 2 provides a procedure to increase the order of convergence of Newton's method by at least one: If $f''(a) \neq 0$ set $m = 2$, else identify the smallest $m > 2$ with $f^{(m-1)}(a) = 0$ and $f^{(m)}(a) \neq 0$. Now form the function $F(x) = f(x)/\sqrt[m]{f'(x)}$, and Newton's method will converge to the root $x = a$ at a rate of $m + 1$ or better. Repeated application will furnish an algorithm to generate an iterative scheme that will converge to a root of $f(x)$ in any desired order, at the expense of the use of higher-order derivatives of f, (and some awkward computations).

Example. Compute $\sqrt[3]{5}$ as a root of $f(x) = x^3 - 5$, and let us assume $x > 0$ throughout. A first application of Theorem 2 with $m = 2$ leads to

$$F_1(x) = \frac{f(x)}{\sqrt{f'(x)}} = \frac{x^3 - 5}{\sqrt{3x^2}} = \frac{1}{\sqrt{3}}\left(x^2 - \frac{5}{x}\right).$$

We may now apply Theorem 2 with $m = 3$ to the function F_1, and after dropping constant factors we obtain

$$F_2(x) = \frac{x^3 - 5}{\sqrt[3]{2x^4 + 5x}}.$$

At first glance it appears that the occurrence of roots in F_1 and F_2 defeats the purpose of finding an algorithm to compute $\sqrt[3]{5}$. But, similar to (5), a look at the resulting Newton iterations shows that computation of roots are not required in the process, as shown in Table 1.

TABLE 1

Function	Newton scheme
$of(x)$	$x_{k+1} = x_k - \frac{x_k^3-5}{3x_k^2}$
$F_1(x)$	$x_{k+1} = x_k - \frac{x_k(x_k^3-5)}{2x_k^3+5}$
$F_2(x)$	$x_{k+1} = x_k - \frac{3x_k(2x_k^3+5)(x_k^3-5)}{10x_k^6+80x_k^3+25}$

In numerical experiments we used $x_0 = 2$ as an initial approximation. The first three iterations for the respective functions with their respective absolute errors are shown below in Table 2.

TABLE 2

	$f(x)$	$F_1(x)$	$F_2(x)$
x_1	1.75	1.714...	1.7103...
error	4.0×10^{-2}	4.3×10^{-3}	3.7×10^{-4}
x_2	1.7108...	1.709975964...	1.7099759466766982...
error	9.1×10^{-4}	1.8×10^{-8}	1.2×10^{-15}
x_3	1.7099764	1.709975946676696989	...
error	4.8×10^{-7}	1.4×10^{-24}	1.5×10^{-60}

In closing it should be pointed out that a variation of Theorem 2 with $m = 1$ can be used to restore quadratic convergence in the case of multiple roots, i.e., if f has a multiple root at $x = a$, then the function $F(x) = f(x)/f'(x)$ has a simple root at $x = a$. See [1] for details.

Appendix. We present the remaining two proofs.

Proof of Theorem 1. Suppose x_k has already been computed. By Taylor's theorem, together with the fact that f and many of its derivatives vanish at $x = a$, there exist constants ξ_0 and ξ_1 between a and x_k such that

$$\begin{aligned} f(x_k) &= f'(a)(x_k - a) + \frac{f^{(m)}(\xi_0)}{m!}(x_k - a)^m \quad \text{and} \\ f'(x_k) &= f'(a) + \frac{f^{(m)}(\xi_1)}{(m-1)!}(x_k - a)^{m-1} \end{aligned}$$

hold. Upon substitution of these expressions into Newton's formula we obtain

$$\begin{aligned} x_{k+1} - a &= x_k - a - \frac{f(x_k)}{f'(x_k)} = \frac{1}{f'(x_k)}\{f'(x_k)(x_k - a) - f(x_k)\} \\ &= \frac{1}{f'(x_k)}\left\{\frac{f^{(m)}(\xi_1)}{(m-1)!} - \frac{f^{(m)}(\xi_0)}{m!}\right\}(x_k - a)^m \\ &= \frac{mf^{(m)}(\xi_1) - f^{(m)}(\xi_0)}{f'(x_k)m!}(x_k - a)^m . \end{aligned}$$

Now we fix a neighborhood $N = [a - l, a + l]$ of a for some suitably small l, so that the inequalities $|f'(x)| \geq c_0 > 0$ and $|f^{(m)}(x)| \leq c_1$ hold true on N for some constants c_0 and c_1. If $x_k \in N$, then

$$\left|\frac{mf^{(m)}(\xi_1) - f^{(m)}(\xi_0)}{f'(x_k)\, m!}\right| \leq \frac{mc_1 + c_1}{m!\, c_0} =: C ,$$

and thus

$$|x_{k+1} - a| \leq C|x_k - a|^m . \tag{2}$$

If necessary, we decrease l such that $l < 1$, and $l^{m-1}C < 1$ are satisfied. Then $x_k \in N$ implies $|x_k - a| \leq l$, and from (2) it follows that $|x_{k+1} - a| \leq C\, l^m \leq l$, i.e., the sequence remains in N and the estimate (2) holds for all subsequent terms of the sequence. □

Proof of Theorem 2. The case $m = 2$ was outlined just before the theorem. We assume $m \geq 3$ and we proceed in two steps.

Step 1. We define the function $g(x) := 1/\sqrt[m]{f'(x)}$. By definition we have $1 = g(x)^m f'(x)$, and implicit differentiation yields

$$0 = mg(x)^{m-1}g'(x)f'(x) + g(x)^m f''(x),$$

and thus

$$0 = mg'(x)f'(x) + g(x)f''(x) . \tag{6}$$

$f''(a) = 0$, together with $f'(a) \neq 0$, implies $g'(a) = 0$. Further differentiation of equation (6) leads to

$$\begin{aligned} 0 &= mg^{(k)}(x)f'(x) + (m(k-1)+1)g^{(k-1)}(x)f''(x) + \cdots \\ &+ \tfrac{(k-1)!}{j!\,(k-j)!}(j + m(k-j))g^{(k-j)}(x)f^{(j+1)}(x) + \cdots \\ &+ (m+k-1)g'(x)f^{(k)}(x) + g(x)f^{(k+1)}(x) . \end{aligned} \tag{7}$$

Since $f''(a) = \cdots = f^{(m-1)}(a) = 0$, equation (7) implies $g^{(k)}(a) = 0$ for $1 \leq k \leq m-2$. For $k = m-1$ and $x = a$ equation (7) reduces to

$$mg^{(m-1)}(a)f'(a) + g(a)f^{(m)}(a) = 0 ,$$

while for $k = m$ and $x = a$ we obtain

$$mg^{(m)}(a)f'(a) + g(a)f^{(m+1)}(a) = 0 .$$

Step 2. We now investigate $F(x) := f(x)g(x)$. Repeated application of the product rule yields

$$\begin{aligned} F'(x) &= f(x)g'(x) + f'(x)g(x) \\ F''(x) &= f(x)g''(x) + 2f'(x)g'(x) + f''(x)g(x) \\ &\vdots \\ F^{(k)}(x) &= f(x)g^{(k)}(x) + kf'(x)g^{(k-1)}(x) + \cdots \\ &+ \binom{k}{j} f^{(j)}(x)g^{(k-j)}(x) + \cdots \\ &+ kf^{(k-1)}(x)g'(x) + f^{(k)}(x)g(x) , \end{aligned}$$

for any function $g(x)$. By definition $F(a) = 0$, and if we choose $g(x)$ as before and use the results from Step 1, we obtain for the derivatives of F at $x = a$

$$
\begin{aligned}
F'(a) &= f'(a)g(a) = f'(a)^{1-1/m} > 0\\
F''(a) &= 2f'(a)g'(a) = 0\\
&\vdots\\
F^{(k)}(a) &= kf'(a)g^{(k-1)}(a) = 0\\
&\vdots\\
F^{(m-1)}(a) &= (m-1)f'(a)g^{(m-2)}(a) = 0\\
F^{(m)}(a) &= mf'(a)g^{(m-1)}(a) + f^{(m)}(a)g(a) = 0\\
F^{(m+1)}(a) &= (m+1)f'(a)g^{(m)}(a) + f^{(m+1)}(a)g(a)\\
&= \left(1 - \frac{m+1}{m}\right) f^{(m+1)}(a)g(a) = -\frac{1}{m}\frac{f^{(m+1)}(a)}{\sqrt[m]{f'(a)}}\,,
\end{aligned}
$$

which proves the theorem. □

REFERENCES

[1] R. L. BURDEN AND J. D. FAIRES, *Numerical Analysis*, 4th ed., Prindle, Weber, Schmidt, Boston, 1989.

[2] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Englewood Cliffs, NJ, 1983.

[3] G. HÄMMERLIN AND K.-H. HOFFMANN, *Numerische Mathematik*, Springer-Verlag, New York, 1989.

[4] J. TODD, *Basic Numerical Mathematics, Vol.* 1, *Numerical Analysis*, Birkhäuser-Verlag, New York, 1979.