Journal of Computational Physics 267 (2014) 139–145

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

# An easy method to accelerate an iterative algebraic equation solver

Jin Yao

*Lawrence Livermore National Laboratory, CA, USA*

ARTICLE INFO

*Article history:*
Received 10 June 2013
Received in revised form 25 January 2014
Accepted 23 February 2014
Available online 5 March 2014

*Keywords:*
Nonlinear
Root-finding
Iterative

ABSTRACT

This article proposes to add a single function calls to an iterative algebraic equation solver with an order $n$ convergence rate, and to raise (for $n > 1$) the order of convergence to $2n - 1$.



## 1. Introduction

Newton's method for finding the root for $f(x) = 0$ is very widely used, both directly and as a conceptual basis for the development of further methods. There is a large literature on schemes to generalize the method to various higher orders. In particular, Kung and Traub [1] demonstrated that an equation solver with $n$ functional calls can achieve an order of convergence $2^{n-1}$. However it is believed that no equation solver that achieves this order has been constructed for $n > 4$.

Many of the existing fast equation solvers are skillfully constructed (for recent examples, see [4–6]) but we consider the simplicity of implementation. In this paper a simple idea is proposed which is to add a single extra function evaluation to an arbitrary one-point iterative equation solver of convergence order $n$, and thereby to accelerate the original scheme to an order of convergence $2n - 1$. Furthermore, it can be shown that with each additional call of the derivatives, the order of convergence is raised by $n - 1$ more (see Appendix A).

## 2. The order of convergence

An iterative equation solver for a set of algebraic equations $\vec{F}(\vec{x}) = 0$ is said to have an order of convergence $n$ when

$$|\vec{x}_{k+1} - \vec{x}_k| = O\left(|\vec{x}_k - \vec{x}_{k-1}|^n\right)$$

at the $k$th iteration. An almost equivalent definition is that

$$\left|\vec{F}(\vec{x}_{k+1})\right| = O\left(|\vec{x}_{k+1} - \vec{x}_k|^n\right)$$

in the case when the Jacobian of the system is *non-zero* at the solution. The order of convergence of an iterative solver is a measurement of how fast it converges to the true solution.

The proposed new scheme accelerates an iterative solver with $n$th-*order* convergence. With a single additional call of the function itself, the order of convergence can be raised from $n$ to $2n - 1$.

http://dx.doi.org/10.1016/j.jcp.2014.02.027
0021-9991/

There are two steps with this method and we demonstrate the procedure here for the case $n = 3$ with the well-known Halley's method [3]. Let $x_k$ be the $k$th estimate for the root. One solves the equation (assuming $f'' \neq 0$)

$$f(x_k) + f'(x_k)\delta + \frac{1}{2} f''(x_k)\delta^2 = 0, \tag{1}$$

and the two roots are explicitly expressed as

$$\delta = -\frac{1}{f''(x_k)}\left(f'(x_k) \pm \sqrt{f'(x_k)^2 - 2f(x_k)f''(x_k)}\right).$$

To recover Newton's method [2] when the quadratic term vanishes, we pick only one root and it can be written as

$$\delta = \frac{\operatorname{sgn}(f'(x_k))}{f''(x_k)}\left(\sqrt{\left(f'(x_k)\right)^2 - 2f(x_k)f''(x_k)} - \left|f'(x_k)\right|\right).$$

The above step uses three functional calls. Note that a Taylor series for $f(x+\delta)$ at $x = x_k$ using Eq. (1) implies

$$f(x_k + \delta) = O\left(\delta^3\right). \tag{2}$$

The next step uses one more function call to gain two more orders of convergence. One adds a term $f(x_k + \delta)$ to Eq. (1), and solves

$$f(x_k + \delta) + f(x_k) + f'(x_k)\Delta + \frac{1}{2} f''(x_k)\Delta^2 = 0. \tag{3}$$

The solution is similar to that obtained in the first step:

$$\Delta = \frac{\operatorname{sgn}(f'(x_k))}{f''(x_k)}\left(\sqrt{\left(f'(x_k)\right)^2 - 2\left(f(x_k+\delta) + f(x_k)\right)f''(x_k)} - \left|f'(x_k)\right|\right).$$

Finally, let $x_{k+1} = x_k + \Delta$ for completion of the current iteration cycle.

One computes only four function values $f(x_k)$, $f'(x_k)$, $f''(x_k)$, and $f(x_k + \delta)$. However, the above scheme is fifth order convergent as shown next.

From a Taylor expansion one obtains

$$f(x_k + \Delta) = f(x_k) + f'(x_k)\Delta + \frac{1}{2} f''(x_k)\Delta^2 + \frac{1}{6} f'''(x_k)\Delta^3 + O\left(\Delta^4\right).$$

The sum of the first three terms in the right-hand side is equal to $-f(x_k + \delta)$ from Eq. (3); thus $f(x_k + \Delta) = -f(x_k + \delta) + f'''(x_k)\Delta^3/6 + O(\Delta^4)$. However, from Eq. (1) and the Taylor expansion of $f(x_k + \delta)$, the above estimate becomes

$$f(x_k + \Delta) = \frac{1}{6} f'''(x_k)\left(\Delta^3 - \delta^3\right) + O\left(\Delta^4 - \delta^4\right) = (\Delta - \delta)O\left(\Delta^2, \delta^2\right). \tag{4}$$

By subtracting Eq. (1) from Eq. (3) one arrives at

$$(\Delta - \delta)\left(f'(x_k) + O(\delta)\right) = -f(x_k + \delta) = O\left(\delta^3\right). \tag{5}$$

It tells us that $\Delta$ and $\delta$ are of the same order and

$$(\Delta - \delta) = O\left(\delta^3\right).$$

One easily sees from Eq. (4) and Eq. (5) that

$$f(x_{k+1}) = f(x_k + \Delta) = O\left(\Delta^5\right).$$

Therefore the method is *fifth-order* convergent; however it employs only *four* function values. The proof above can be generalized for arbitrary $n$.

If $f'(x) = 0$ at the solution, Newton's method either fails or converges slowly. The rapidly-converging scheme described above is more stable. However, the order of convergence is reduced from *five* to *four* because when $f' = 0$, Eq. (5) gives $(\delta - \Delta) = O(\delta^2)$ instead of $O(\delta^3)$. In practice if $x_k$ is not close to the solution, the term under the square root $(f'(x_k))^2 - 2f(x_k)f''(x_k)$ can become negative and break the iteration. In this case this term can be set to zero to keep the computation going.

If $f'$ is finite at the solution, because $f \to 0$ when $x_k$ is close to the solution, $x^*$, the term in the square root will be non-negative when sufficiently close to convergence; if $f' = 0$ at the solution, setting this term to zero (if it becomes negative) would give $\delta = -f'(x_k)/f''(x_k)$, which is similar to a Newton-method by L'Hospital's rule.