Available online at www.sciencedirect.com
ScienceDirect
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
ELSEVIER
Journal of Computational and Applied Mathematics 218 (2008) 350–363
www.elsevier.com/locate/cam

Simple geometry facilitates iterative solution of a nonlinear equation via a special transformation to accelerate convergence to third order

M. Çetin Koçak*

Chemical Engineering Department, Engineering Faculty, Ankara University, 06100 Ankara, Turkey

Received 22 August 2006; received in revised form 24 February 2007

Abstract

Direct substitution $x_{k+1} = g(x_k)$ generally represents iterative techniques for locating a root z of a nonlinear equation $f(x)$. At the solution, $f(z) = 0$ and $g(z) = z$. Efforts continue worldwide both to improve old iterators and create new ones. This is a study of convergence acceleration by generating secondary solvers through the transformation $g_m(x) = (g(x) - m(x)x)/(1 - m(x))$ or, equivalently, through partial substitution $g_{mps}(x) = x + G(x)(g - x)$, $G(x) = 1/(1 - m(x))$. As a matter of fact, $g_m(x) \equiv g_{mps}(x)$ is the point of intersection of a linearised g with the $g = x$ line. Aitken's and Wegstein's accelerators are special cases of g_m. Simple geometry suggests that $m(x) = (g'(x) + g'(z))/2$ is a good approximation for the ideal slope of the linearised g. Indeed, this renders a third-order g_m. The pertinent asymptotic error constant has been determined. The theoretical background covers a critical review of several partial substitution variants of the well-known Newton's method, including third-order Halley's and Chebyshev's solvers. The new technique is illustrated using first-, second-, and third-order primaries. A flexible algorithm is added to facilitate applications to any solver. The transformed Newton's method is identical to Halley's. The use of $m(x) = (g'(x) + g'(z))/2$ thus obviates the requirement for the second derivative of $f(x)$. Comparison and combination with Halley's and Chebyshev's solvers are provided. Numerical results are from the square root and cube root examples.

MSC: 37M05; 65B99; 65H05

Keywords: Algebraic equations solvers; Iterative methods; Fixed-point iterations; Simulation; Convergence order; Direct substitution; Partial substitution; Halley's method; Newton's method; Acceleration of iteration

1. Introduction

Efficient solution techniques are required for nonlinear equations which partake of scientific, engineering, economy, and various other models. Simulation applications, especially dynamic ones, may need multiple runs each demanding thousands of times zeroes of algebraic equations coupled to differential equations. Often, either lack or intractability of an analytical solution directs one to harness an iterative technique for this task and face possibilities of slow convergence, non-convergence and divergence, in other words, inefficiency or failure.

Let z be a zero of an arbitrary nonlinear function $f(x)$, that is, let $f(z) = 0$. Further, appoint k to count the iterations. A fixed-point iteration method starts from one or more guessed x values, thereafter repeatedly uses direct substitution

* Tel.: +90 312 2126720.
E-mail address: kocak@eng.ankara.edu.tr.

0377-0427/$ - see front matter
doi:10.1016/j.cam.2007.02.036

$x_{k+1} = g(x_k)$ until it reaches a point where two consecutive x values coincide within a pre-specified tolerance, and reports this point as z.

Let ε_k denote the error in the kth iterate, that is, let $\varepsilon_k = x_k - z$. By definition, if $\lim_{k\to+\infty}(|\varepsilon_{k+1}|/|\varepsilon_k|^n) = c_{as}$, then n and c_{as} are the convergence order and the asymptotic error constant. "*Linear*" or "*first-order*" methods are those where $n = 1$, $g'(z) \neq 0$ and so $|\varepsilon_{k+1}|$ is proportional to $|\varepsilon_k|$ in the neighbourhood of z. For $n = 2$, the only requirement is $g'(z) = 0$ whereas for $n > 2$, this becomes $g'(z) = g''(z) = \cdots = g^{(n-1)}(z)$ and $g^{(n)}(z) \neq 0$.

Remark 1. nth order solvers are a subset of $(n-1)$th order solvers. Higher n may mean fewer but not necessarily cheaper iterations. Irrespective of n, there is always a risk of divergence if not started close enough to z. Iteration equations obtained from a rearrangement of $f(x) = 0$ are mostly linear. (See illustrations 1–3 presented later.)

Beyond this point in the text, when functions are written without an argument the latter is x. Newton's popular technique [1,3–5,13] is a computationally simple, one-point method without memory: $g_N = x - f/f'$. It is a piecewise linearisation of f since it extends the current tangent to intersect the x-axis and suggests this value as the next approximation to z. For simple roots this method is second order. Repetition of z demotes convergence from quadratic to *superlinear* or *geometrical* and that slows down the iteration process. If r is the multiplicity of z, then $g'(z) = (r-1)/r \neq 0$ and $g_{Nr} = x - rf/f'$ restores second-order convergence [5].

Partial substitution is an attempt to improve convergence of a given solver by insertion of a variable gain G into the direct substitution formula so that it becomes $g_{ps} = x + G(g - x)$. Applied to Newton's scheme, partial substitution gives $g_{Nps} = x - Gf/f'$.

Remark 2. $g_{Nr} = x - rf/f'$ is a g_{Nps} with a *fixed* $G = r$.

Chebychev's ($g_C = x - f/f' - f^2 f''/(2f'^3)$) and Halley's ($g_H = x - ff'/(f'^2 - 0.5ff'')$) methods are two well-known solvers that are of third order for simple roots. They are, like so many others, g_{Nps} variants (see Section 2). Needless to say, both g_C and g_H are more involved and demanding than g_N since they require f'' in addition to f and f'.

Efforts continue worldwide both to boost existing iterative solvers and develop new ones. The new transformation of this research has been applied to accelerate linear solvers besides higher order ones like g_N, g_{Nr}, g_C, and g_H. Now, g_N and its g_{Nps} variants (including g_{Nr}, g_C, and g_H) belong to set of solvers in the form $g_u = x + fu$ where u is a weight function. With an obvious notation, $u_N = -1/f'$ and $u_{Nps} = -G/f'$.

2. Some solvers of the type $g_u = x + fu$

Direct differentiation of $g_u = x + fu$ with respect to x gives

$$g_u' = 1 + f'u + fu', \quad g_u'' = f''u + 2f'u' + fu'', \quad g''' = f'''u + 3f''u' + 3f'u'' + fu'''.$$

As x goes to z the terms containing f disappear because $f(z) = 0$ and so these derivatives tend towards

$$g_u'(z) = 1 + f'(z)u(z), \quad g_u''(z) = f''(z)u(z) + 2f'(z)u'(z), \quad g_u'''(z) = f'''(z)u(z) + 3f''(z)u'(z) + 3f'(z)u''(z).$$

Equating $g'(z)$, $g''(z)$, and $g'''(z)$ to zero and rearranging, one obtains the first three rungs up the ladder of convergence order for g_u methods:

$$g_u'(z) = 0 \Rightarrow \quad u(z) = -1/f'(z), \tag{1a}$$

$$g_u''(z) = 0 \Rightarrow \quad u'(z) = -0.5f''(z)u(z)/f'(z) = 0.5f''(z)/f'^2(z), \tag{1b}$$

$$g_u'''(z) = 0 \Rightarrow \quad u''(z) = -(f'''(z)u(z) + 3f''(z)u'(z))/(3f'(z)) = \frac{f'''(z)}{3f'^2(z)} - \frac{f''^2(z)}{2f'^3(z)}. \tag{1c}$$

Remark 3. The implicit condition in (1) that $f'(z) \neq 0$ means z must be simple root $(r = 1)$.

Violation of (1a) is enough to make g_u a first-order process. If u satisfies only (1a), then g_u is a second-order process. If u satisfies (1a) and (1b) but fails (1c), then $n = 3$. If u fulfils (1a), (1b), and (1c), then n is (at least) 4. It is relatively