An Efficient One-Point Extrapolation Method
for Linear Convergence*

By Richard F. King

Abstract. For iteration sequences otherwise converging linearly, the proposed one-point extrapolation method attains a convergence rate and efficiency of 1.618. This is accomplished by retaining an estimate of the linear coefficient from the previous step and using the estimate to extrapolate. For linear convergence problems, the classical Aitken-Steffensen δ^2-process has an efficiency of just $\sqrt{2}$, while a recently proposed fourth-order method reaches an efficiency of 1.587. Not only is the method presented here more efficient, but it is also quite straightforward. Examples given are for Newton's method in finding multiple polynomial roots and for locating a fixed point of a nonlinear function.

1. Introduction. We are concerned with extrapolation methods for improving the convergence rate of a sequence $x_0, x_1, \ldots \longrightarrow \alpha$ exhibiting only linear convergence. Presume that the sequence is generated by a one-point iteration function ϕ from an initial point x_0:

$$(1.1) \qquad x_{n+1} = \phi(x_n), \quad n = 0, 1, 2, \ldots.$$

The error equation

$$(1.2) \qquad \epsilon_{n+1} = K\epsilon_n + L\epsilon_n^2 + M\epsilon_n^3 + \cdots,$$

derived for error $\epsilon_n = x_n - \alpha$ by expanding $\phi(x_n)$ in a Taylor series about α, then has $0 < |K| < 1$. Here we have taken $K = \phi'(\alpha)$, $L = \phi''(\alpha)/2!$, and $M = \phi'''(\alpha)/3!$.

The 'δ-process', developed by Aitken [1] and applied to ϕ by Steffensen [7] (see [3, pp. 135–139] or [6, Appendix E]), is one such extrapolation method. It may be derived by applying a single step of the secant method,

$$(1.3) \qquad \bar{x}_{n+2} = x_{n+1} - (x_n - x_{n+1}) \frac{g(x_{n+1})}{g(x_n) - g(x_{n+1})},$$

to the function $g = x - \phi$, remembering that $\phi(x_0) = x_1$ and $\phi(x_1) = x_2$. The resulting extrapolation, used after every two linear-error substeps, is

$$(1.4) \qquad \bar{x}_2 = \frac{x_0 x_2 - x_1^2}{x_0 - 2x_1 + x_2},$$

where here, and henceforth in this section, we suppress n in the subscripts. Now use (1.2) and (1.4) to get the error equation

$$(1.5) \qquad \bar{\epsilon}_2 = \frac{\epsilon_0 \epsilon_2 - \epsilon_1^2}{\epsilon_0 - 2\epsilon_1 + \epsilon_2} \cong \left(\frac{-LK}{1 - K} \right) \epsilon_0^2,$$

for the error $\bar{\epsilon}_2 = \bar{x}_2 - \alpha$. Extrapolated iterate $\bar{x}_2$ becomes a new x_0 for two more linear-error substeps and another extrapolation, and so forth. Thus, the procedure has convergence of order two; its efficiency in the sense of Traub [8, p. 263] is $\sqrt{2}$, because two function evaluations are required for each step. The δ^2-process is a two-point method without memory [8, p. 8].

Another extrapolation method for a linearly converging sequence is described in [5]. It is a fourth-order scheme requiring just three evaluations of ϕ per step, and so enjoys an improved efficiency of $4^{1/3} = 1.587$ versus 1.414 for Aitken-Steffensen extrapolation. The method may be described as follows:

$$(\text{i}) \quad x_1 = \phi(x_0) \; ,$$

$$(\text{ii}) \quad x_2 = \phi(x_1) \; ,$$

$$(1.6) \qquad (\text{iii}) \quad \bar{x}_2 = x_1 - \left(\frac{1}{1 - K_1} \right) (x_1 - x_2) \; ,$$

$$(\text{iv}) \quad x_3 = \phi(\bar{x}_2) \; ,$$

$$(\text{v}) \quad \bar{x}_3 = \bar{x}_2 - \left(\frac{1}{1 - \hat{K}} \right) (\bar{x}_2 - x_3) \; ,$$

where

$$(1.7) \quad K_1 = \frac{x_2 - x_1}{x_1 - x_0} \doteq K + L(K + 1)\epsilon_0 + [L^2 + M(K^2 + K + 1)] \epsilon_0^2 \; ,$$

$$(1.8) \quad K^* = \frac{x_3 - x_2}{x_2 - x_1} \doteq K + LK\epsilon_0 + [1 - K]^{-1}[L^2(-2K + 1) + M(-K^3 + K^2)] \epsilon_0^2 \; ,$$

$$(1.9) \quad \hat{K} = K^*(1 + K^* - K_1) \doteq K + [1 - K]^{-1}[L^2(-3K + 1) + M(K^2 - K)] \epsilon_0^2 \; .$$

Linear-error substeps (i) and (ii), together with extrapolation (iii), constitute one step of the δ^2-process. A third linear-error substep (iv) (applying ϕ to $\bar{x}_2$) and final extrapolation (v) complete the procedure. Note that both K_1 and K^* are merely first-order approximations to K but combine as in (1.9) to form the second-order approximation $\hat{K}$. It is the second-order accuracy of the approximation $\hat{K}$ to K in extrapolation (v) that gives the method its fourth-order convergence. The iterate $\bar{x}_3$ at the end of each step becomes x_0 for the next step. For this method the error equation turns out to be

$$(1.10) \qquad \bar{\epsilon}_3 = [1 - K]^{-3}[L^3(-2K^2 + K) + ML(K^3 - K^2)] \epsilon_0^4.$$

This is a three-point method without memory.

In the next section we develop a one-point extrapolation method with memory. An estimate of K based on two current and two intermediate points is used to extrapolate after each linear-error step. The result is a method that is more efficient than

either of the methods heretofore discussed, and that is incidentally quite straight-forward.

2. An Extrapolation Method With Memory. First observe that the extrapolation (v) of (1.6) is possible after just one new function evaluation (iv) only because an estimate $\hat{K}$ of K is available from current and previous substeps. We use this idea as the basis of the proposed method: each step consists of a linear-error substep followed by an extrapolation requiring information retained not only from the present step, but also from the preceding one. If we define

$$(2.1) \qquad \widetilde{K}_{n+1} = \frac{x_{n+2} - x_{n+1}}{\widetilde{x}_{n+1} - \widetilde{x}_n} \,,$$

then one step of the method may be written as

$$(2.2) \qquad \begin{array}{l} \text{(i)} \quad x_{n+2} = \phi(\widetilde{x}_{n+1}) \,, \\[2mm] \text{(ii)} \quad \widetilde{x}_{n+2} = \widetilde{x}_{n+1} - \left(\frac{1}{1 - \widetilde{K}_{n+1}}\right) (\widetilde{x}_{n+1} - x_{n+2}) \,. \end{array}$$

Start the process by evaluating $x_1 = \phi(x_0)$ and $x_2 = \phi(x_1)$, and setting $\widetilde{x}_0 \equiv x_0$ and $\widetilde{x}_1 \equiv x_1$. Then $\widetilde{K}_1 = (x_2 - x_1)/(x_1 - x_0)$, so the first extrapolation is simply that of Aitken-Steffensen. Our new sequence of iterates is $\{\widetilde{x}_n\}, n = 0, 1, 2, \dots$.

Another way to derive the scheme, analogous to (1.3) for the δ^2-process, is to apply the secant method to the function $h(\widetilde{x}) = \widetilde{x} - \phi(\widetilde{x})$. Thus

$$(2.3) \qquad \widetilde{x}_{n+2} = \widetilde{x}_{n+1} - (\widetilde{x}_n - \widetilde{x}_{n+1}) \left(\frac{h(\widetilde{x}_{n+1})}{h(\widetilde{x}_n) - h(\widetilde{x}_{n+1})}\right),$$

and (2.3) is seen to be equivalent to (2.2).

To get the error equation, we must find out how good $\widetilde{K}_{n+1}$ is as an estimate of K. From (1.1), (1.2), and (i) of (2.2), we can write

$$(2.4) \qquad \epsilon_{n+2} \doteq K\widetilde{\epsilon}_{n+1} + L\widetilde{\epsilon}_{n+1}^2, \qquad \epsilon_{n+1} \doteq K\widetilde{\epsilon}_n + L\widetilde{\epsilon}_n^2.$$

Then from (2.1) and (2.4) it follows that

$$(2.5) \qquad \widetilde{K}_{n+1} = \frac{\epsilon_{n+2} - \epsilon_{n+1}}{\widetilde{\epsilon}_{n+1} - \widetilde{\epsilon}_n} \doteq K + L\widetilde{\epsilon}_n,$$

since ultimately $|\widetilde{\epsilon}_{n+1}| \ll |\widetilde{\epsilon}_n|$. Finally, the error equation can be calculated from (ii) of (2.2), using (2.4) and (2.5), as

$$(2.6) \qquad \widetilde{\epsilon}_{n+2} \cong \left(\frac{-L}{1 - K}\right) \widetilde{\epsilon}_n \widetilde{\epsilon}_{n+1}.$$

Thus, the error equation has the same form as that of the classical secant method for a simple root. Consequently, the method has a convergence rate of $(1 + \sqrt{5})/2 \doteq 1.618$. Since only one evaluation of ϕ is required each step, it also has an efficiency of 1.618, as against 1.587 for the method of [5] and 1.414 for the δ^2-process. It may be classified as a one-point method with memory.

In none of the three schemes discussed do succeeding elements of the sequence defined by (1.1) need to be evaluated once extrapolation has begun. For instead the

function ϕ is applied to more accurate estimates of α than the terms of the original sequence.

3. Examples. a. The first two examples are for Newton's method,

$$(3.1) \qquad x_{n+1} = \phi(x_n) = x_n - \frac{f(x_n)}{f'(x_n)} \;,$$

applied to a nonlinear function f with a multiple root at α of multiplicity $m \neq 1$. It has been shown in [5] that for derivative ratios

$$(3.2) \qquad R = \frac{f^{(m+1)}(\alpha)}{f^{(m)}(\alpha)} \;, \qquad S = \frac{f^{(m+2)}(\alpha)}{f^{(m)}(\alpha)} \;,$$

the coefficients in (1.2) are given by

$$(3.3) \quad K = \frac{m-1}{m} \;, \quad L = \frac{R}{m^2(m+1)} \;, \quad M = \frac{2S}{m^2(m+1)(m+2)} - \frac{R^2}{m^3(m+1)} \;.$$

Tables 1 and 2 list the results of computations done on an IBM 370 in quadruple precision for functions $f = (x-1)^2 \tan(\pi x/4)$ and $f = x \sin([x-1]^4)$, respectively. Both the actual error $\epsilon = x - \alpha$ and the error estimate computed from (2.6), as well as the function value f, are given for the proposed method. Since $m = 1/(1-K)$, from (3.3), it follows that $\widetilde{m} = 1/(1 - \widetilde{K})$ is an estimate of m; both $\widetilde{K}$ (from (2.1)) and $\widetilde{m}$ are shown in the tables. For comparison, results using the δ^2-process are also included. Extrapolation substeps have a T (for tilde) following the number n of the substep in the first column.

TABLE 1

f			α	x_0	m	K	L	M
$(x-1)^2 \tan(\pi x/4)$			1	.5	2	1/2	$\pi/8$	$-\pi^2/32$
$\widetilde{\epsilon}_{n+2} \stackrel{=}{\approx} [-L][1-K]^{-1} \; \widetilde{\epsilon}_n \; \widetilde{\epsilon}_{n+1} = [-\pi/4] \; \widetilde{\epsilon}_n \; \widetilde{\epsilon}_{n+1}$								
n	$x - \alpha$	f	$[-\pi/4]\widetilde{\epsilon}_n \; \widetilde{\epsilon}_{n+1}$	$\widetilde{K}$	$\widetilde{m}$		δ^2-process $x-\alpha$	
0	-.500000	.103553					-.500000	
1	.622531(-1)	.427423(-2)					.622531(-1)	
2	.325841(-1)	.111751(-2)					.325841(-1)	
2T	.340712(-1)	.122470(-2)					.340712(-1)	
3	.174802(-1)	.314065(-3)		.535944	2.154914		.174802(-1)	
3T	-.168097(-2)	.281821(-5)	-.166586(-2)					
4	-.839374(-3)	.703620(-6)		.512405	2.050881		.885852(-2)	
4T	.450433(-4)	.202904(-8)	.449819(-4)				-.468967(-3)	
5	.225224(-4)	.507278(-9)		.499357	1.997430		-.234397(-3)	
5T	.594677(-7)	.353641(-14)	.594675(-7)					
6	.297339(-7)	.884103(-15)		.500018	2.000071		-.117177(-3)	
6T	-.210378(-11)	.442591(-23)	-.210378(-11)				-.863344(-7)	
7	-.105189(-11)	.110648(-23)		.50000002	2.00000009		-.431672(-7)	
7T	.982590(-19)	.965483(-38)	.982590(-19)					

Inasmuch as both f and f' in the function ϕ must be evaluated each substep for Newton's method, the effective efficiency here for the δ^2-process is really only $2^{1/4} \doteq 1.189$; see Van de Vel [9]. Effective efficiency for the method of [5] is correspondingly $4^{1/6} \doteq 1.260$; this matches the efficiency of the nonextrapolation root-finding procedure of Esser; see [2]. For the proposed method of (2.1) and (2.2), the effective efficiency with Newton's method is $(1.618)^{1/2} \doteq 1.272$; this efficiency is also that of the procedure outlined in [4], in which the secant method itself is applied to a function

analogous to f/f', namely, to a related function having a simple zero at α.

TABLE 2

f			α	x_0	m	K	L	M
x sin([x-1]*)			1	.5	4	3/4	1/16	-5/64
$\tilde{\epsilon}_{n+2} \cong [-L][1-K]^{-1} \ \tilde{\epsilon}_n \ \tilde{\epsilon}_{n+1} = [-1/4] \ \tilde{\epsilon}_n \ \tilde{\epsilon}_{n+1}$								
n	x-α	f	$[-1/4]\tilde{\epsilon}_n \ \tilde{\epsilon}_{n+1}$	$\tilde{K}$	$\tilde{m}$	δ^2-process x-α		
0	-.500000	.312297(-1)				-.500000		
1	-.333043	.820521(-2)				-.333043		
2	-.237900	.244113(-2)				-.237900		
2T	-.111849	.139001(-3)				-.111849		
3	-.829778(-1)	.434739(-4)		.700391	3.337685	-.829778(-1)		
3T	-.154860(-1)	.566206(-7)	-.931265(-2)					
4	-.115992(-1)	.178914(-7)		.740726	3.856921	-.617533(-1)		
4T	-.494979(-3)	.599975(-13)	-.433023(-3)			-.284075(-2)		
5	-.371219(-3)	.189828(-13)		.748981	3.983769	-.213006(-2)		
5T	-.194741(-5)	.143823(-22)	-.191631(-5)					
6	-.146056(-5)	.455066(-23)		.749969	3.999503	-.159726(-2)		
6T	-.241102(-9)	.308149(-32)	-.240982(-9)			-.152028(-5)		

b. The last example is for a direct application of the iteration $x_{n+1} = \phi(x_n)$ of (1.1) in finding that fixed point of

$$(3.4) \qquad \phi = \frac{e^{x-1} + 1}{2}$$

at $\alpha = 1$. An initial estimate of $x_0 = .5$ is used. (The other fixed point, at $x \doteq 2.256$, does not concern us.) In this case the coefficients in (1.2) are $K = 1/2, L = 1/4$, and $M = 1/12$. We can think of the problem as that of finding a root of the nonlinear function $f = x - \phi$.

Quadruple-precision results are shown in Table 3. Again the actual error $\epsilon = x - \alpha$, the function value f, and the error estimate from (2.6) are given, as is $\widetilde{K}$ from (2.1). A T denotes an extrapolation step. Corresponding results for the δ^2-process are also listed.

TABLE 3

ϕ		f		α	x_0	K	L	M
[e^{x-1}+1]/2		x-ϕ(x)		1	.5	1/2	1/4	1/12
$\tilde{\epsilon}_{n+2} \cong [-L][1-K]^{-1} \ \tilde{\epsilon}_n \ \tilde{\epsilon}_{n+1} = [-1/2] \ \tilde{\epsilon}_n \ \tilde{\epsilon}_{n+1}$								
n	x-α	f	$[-1/2]\tilde{\epsilon}_n \ \tilde{\epsilon}_{n+1}$	$\tilde{K}$		δ^2-process x-α		
0	-.500000	-.303265				-.500000		
1	-.196735	-.107439				-.196735		
2	-.892957(-1)	-.465833(-1)				-.892957(-1)		
2T	-.303500(-1)	-.154030(-1)				-.303500(-1)		
3	-.149470(-1)	-.752909(-2)		.446848		-.149470(-1)		
3T	-.250417(-2)	-.125365(-2)	-.298545(-2)					
4	-.125052(-2)	-.625651(-3)		.491869		-.741794(-2)		
4T	-.369864(-4)	-.184935(-4)	-.380009(-4)			-.218535(-3)		
5	-.184929(-4)	-.924652(-5)		.499365		-.109255(-3)		
5T	-.462123(-7)	-.231062(-7)	-.463102(-7)					
6	-.231062(-7)	-.115531(-7)		.4999907		-.546247(-4)		
6T	-.854588(-12)	-.427294(-12)	-.854614(-12)			-.119348(-7)		
7	-.427294(-12)	-.213647(-12)		.499999988		-.596740(-8)		
7T	-.197462(-19)	-.987312(-20)	-.197462(-19)					

Note that here the iteration of (1.1) is known in advance to converge just linearly—if it converges at all—because $\phi' = [\exp(x - 1)]/2 > 0$ everywhere. Nevertheless, the efficiency (1.618) of the proposed method for this problem is as high as that of the secant method for finding a simple root of the nonlinear function f.

4. Acknowledgement. The author extends thanks to D. L. Phillips for carefully reading the manuscript and offering useful suggestions for its improvement.

Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439

1. A. C. AITKEN, "On Bernoulli's numerical solution of algebraic equations," *Proc. Roy. Soc. Edinburgh*, v. 46, 1926, pp. 289–305.

2. H. ESSER, "Eine stets quadratisch konvergente Modifikation des Steffensen-Verfahrens," *Computing*, v. 14, 1975, pp. 367–369.

3. A. S. HOUSEHOLDER, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, New York, 1970.

4. R. F. KING, "A secant method for multiple roots," *BIT*, v. 17, 1977, pp. 321–328.

5. R. F. KING, "An extrapolation method of order four for linear sequences," *SIAM J. Numer. Anal.*, v. 16, 1979, pp. 719–725.

6. A. M. OSTROWSKI, *Solution of Equations and Systems of Equations*, 2nd ed., Academic Press, New York, 1966.

7. J. F. STEFFENSEN, "Remarks on iteration," *Skandinavisk Aktuarietidskrift*, v. 16, 1933, pp. 64–72.

8. J. F. TRAUB, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, N.J., 1964.

9. H. VAN DE VEL, "A method for computing a root of a single nonlinear equation, including its multiplicity," *Computing*, v. 14, 1975, pp. 167–171.