Anderson-Björck for Linear Sequences*

By Richard F. King

Abstract. The proposed one-point method for finding the limit of a slowly converging linear sequence features an Anderson-Björck extrapolation step that had previously been applied to the Regula Falsi problem. Convergence is of order 1.839 as compared to $\sqrt{2}$ for the well-known Aitken-Steffensen δ^2 -process, and to 1.618 for another one-point extrapolation procedure of King. There are examples for computing a polynomial's multiple root with Newton's method and for finding a fixed point of a nonlinear function.

1. Introduction. Let us suppose that we have a stationary, one-point generating function ϕ for a sequence $x_0, x_1, \dots$, but that convergence to the limit α is only linear. That is, starting from a given x_0 ,

$$(1.1) \quad x_{n+1} = \phi(x_n) \equiv \phi_n, \quad n = 0, 1, 2, \dots,$$

and the error $\epsilon_n = x_n - \alpha$ satisfies

$$(1.2) \quad \epsilon_{n+1} = K\epsilon_n + L\epsilon_n^2 + M\epsilon_n^3 + N\epsilon_n^4 + \dots,$$

with $0 < |K| < 1$. The error equation (1.2) for ϕ we get by expanding $\phi(x_n)$ in a Taylor series about α , noting that $\phi(\alpha) = \alpha$, and setting $K = \phi'(\alpha)$, $L = \phi''(\alpha)/2!$, $M = \phi'''(\alpha)/3!$, and $N = \phi^{iv}(\alpha)/4!$. The function ϕ is called stationary because it is independent of n .

Several extrapolation methods for improving the convergence rate are known. We shall review two such methods, and then develop an even faster extrapolation procedure—one based on the Anderson-Björck step introduced and used in [2] for finding a bracketed simple root of a nonlinear equation.

Each of the extrapolation methods may be derived by applying the secant method to a suitable ϕ -related function having a simple zero at α . Since the secant method is hyperlinear for a simple α , so will the resulting extrapolation methods be hyperlinear.

In the classical δ^2 -process, developed by Aitken [1] and applied to ϕ by Steffensen [10] (see [4, pp. 135–139] or [9, Appendix E]), the related function is $g = x - \phi$. The point α is a simple root for g because $g(\alpha) = \alpha - \phi(\alpha) = 0$, and $g'(\alpha) = 1 - \phi'(\alpha) = 1 - K \neq 0$. Each step of the δ^2 -process goes as follows: from x_n take two linear-error substeps, $x_{n+1} = \phi(x_n)$ and $x_{n+2} = \phi(x_{n+1})$. The extrapolation substep

Received June 24, 1982.

1980 *Mathematics Subject Classification*. Primary 65B99, 65H05.

Key words and phrases. Linear convergence, extrapolation, Aitken's δ^2 -process, Steffensen, Anderson-Björck, efficiency, nonlinear equation, order of convergence, Regula Falsi, linear sequence.

*Work performed under the auspices of the U. S. Department of Energy.

©1983 American Mathematical Society
0025-5718/83 \$1.00 + \$.25 per page

then consists of one application of the secant method

$$(1.3) \quad \bar{x}_{n+2} = x_{n+1} - (x_n - x_{n+1}) \frac{g_{n+1}}{g_n - g_{n+1}}$$

to g , wherein $g_n \equiv g(x_n) = x_n - x_{n+1}$ and $g_{n+1} \equiv g(x_{n+1}) = x_{n+1} - x_{n+2}$. Note that the extrapolation itself does not require any evaluations of ϕ . The corresponding asymptotic error equation

$$(1.4) \quad \bar{\epsilon}_{n+2} = \frac{\epsilon_n \epsilon_{n+2} - \epsilon_{n+1}^2}{\epsilon_n - 2\epsilon_{n+1} + \epsilon_{n+2}} \cong \left(\frac{-LK}{1-K} \right) \epsilon_n^2$$

for error $\bar{\epsilon}_{n+2} = \bar{x}_{n+2} - \alpha$ comes out of (1.2) and (1.3). Next, point $\bar{x}_{n+2}$ becomes a new x_n for two more linear-error substeps and another extrapolation, and so on. From (1.4), the whole δ^2 -process has second-order convergence; its efficiency in the sense of Traub [11, p. 263] is $\sqrt{2}$, because ϕ must be evaluated twice per step. Again following Traub [11, p. 8], the δ^2 -process is a two-point method without memory.

But we can attain a higher convergence rate merely by retaining, at each step, information held over from the previous step. This idea is incorporated (with a change of notation) into the one-point extrapolation method with memory described in [7], to wit: define $\phi_n = \phi(x_n)$, and get started from x_0 by taking $x_1 = \phi_0$. For each and every subsequent step, apply a secant-method step to the related function $g = x - \phi$. Thus the method is

$$(1.5) \quad x_{n+2} = x_{n+1} - (x_n - x_{n+1}) \frac{g_{n+1}}{g_n - g_{n+1}},$$

starting at $g_0 \equiv g(x_0) = x_0 - \phi_0$ and $g_1 \equiv g(x_1) = x_1 - \phi_1$. Since ultimately $|\epsilon_{n+1}| \ll |\epsilon_n|$, the error equation for the one-point method of [7] turns out to be the following (from (1.5) and the error expression (1.2) for ϕ):

$$(1.6) \quad \epsilon_{n+2} \cong \left(\frac{-LK}{1-K} \right) \epsilon_n \epsilon_{n+1}.$$

That is to say, the error equation is of the same form as that of the usual secant method for a simple root. Thus the method has a convergence rate of $(1 + \sqrt{5})/2 \doteq 1.618$. Furthermore the scheme also has an efficiency of 1.618 because only one new ϕ -value needs to be calculated each step. This compares with 1.414 for the δ^2 -process.

In passing, we mention several other procedures that, while not as efficient as the method to be developed, are nevertheless hyperlinear: the extrapolation methods of Van de Vel [12] (efficiency 1.414) and of King [6] (1.587), in which the linear error term for ϕ is effectively subtracted out; the nonextrapolation methods of Esser [3] (1.587) and of King [5] (1.618), in which ϕ is replaced by a hyperlinear generating function using divided differences and analogous to $x - (x - \phi)/(1 - \phi)$.

But now, on to the proposed extrapolation procedure.

2. A New Iteration Procedure Based on the Anderson-Björck Extrapolation Step.

The idea in an Anderson-Björck step is (i) to fit a parabola $\tilde{g}_{n+2}$ to three points (x_n, g_n) , (x_{n+1}, g_{n+1}) , (x_{n+2}, g_{n+2}) of a function g whose zero is sought, (ii) to compute the derivative $\tilde{g}'_{n+2}$ to $\tilde{g}_{n+2}$ at x_{n+2} , and then (iii) to take a Newton-like

step from x_{n+2} to a new x_{n+3} , using $g_{n+2} \equiv \tilde{g}_{n+2}$ and $\tilde{g}'_{n+2}$:

$$(2.1) \quad x_{n+3} = x_{n+2} - \frac{g_{n+2}}{\tilde{g}'_{n+2}}.$$

It turns out that $\tilde{g}'_{n+2}$ can be written (and computed) in terms of divided differences of g as

$$(2.2) \quad \tilde{g}'_{n+2} = g[x_{n+2}, x_{n+1}] + g[x_{n+2}, x_n] - g[x_n, x_{n+1}],$$

where $g[x_j, x_k] = (g_k - g_j)/(x_k - x_j)$.

In our procedure we take g to be the ϕ -related function $g = x - \phi$. Starting from a given x_0 , we (1) compute $\phi(x_0) \equiv \phi_0$ and set $g_0 = x_0 - \phi_0$. Now (2) take $x_1 = \phi_0$, compute $\phi(x_1) \equiv \phi_1$, and set $g_1 = x_1 - \phi_1$. At this point we could have chosen to repeat the cycle, i.e., (3*) take $x_2 = \phi_1$, compute ϕ_2 , and set $g_2 = x_2 - \phi_2$. Finally (4) find x_3 and succeeding points by means of the Anderson-Björck step (2.1), and at each point set the new $g_{n+3} = x_{n+3} - \phi_{n+3}$.

But after step (2) we already have enough information to apply the δ^2 -process to points $P_0 = (x_0, g_0)$ and $P_1 = (x_1, g_1)$. This we choose to do because it accelerates the convergence process. Consequently, instead of (3*) in our procedure we substitute the following: Step (3) take x_2 to be the δ^2 -extrapolant of (x_0, g_0) and (x_1, g_1) , compute $\phi(x_2) \equiv \phi_2$, and set $g_2 = x_2 - \phi_2$. It is easy to show that we can accomplish the programming of Step (3) quite simply by setting (in sequence) $(x_2, g_2) = (x_1, g_1)$, $(x_1, g_1) = (x_0, g_0)$, $(x_0, g_0) = (2x_1 - x_2, 2g_1 - g_2)$, and then using the general step (2.1). (In this case the parabola $\tilde{g}_2$ degenerates into the straight line through P_0 and P_1 .)

We know that g has a simple root at α (because $g(\alpha) = \alpha - \phi(\alpha) = 0$ and $g'(\alpha) = 1 - \phi'(\alpha) = 1 - K \neq 0$). Furthermore, $g''(\alpha) = -2L$, $g'''(\alpha) = -6M$, and $g^{iv}(\alpha) = -24N$. It can then readily be shown from (1.2), (2.1), and (2.2) that

$$(2.3) \quad \epsilon_{n+3} = \frac{M}{1-K} \epsilon_n \epsilon_{n+1} \epsilon_{n+2} - \frac{L}{1-K} \epsilon_{n+2}^2 + \frac{N}{1-K} \epsilon_n^2 \epsilon_{n+1} \epsilon_{n+2} + \cdots.$$

Asymptotically the second and third terms are negligible, so the error equation for our procedure is

$$(2.4) \quad \epsilon_{n+3} \cong \frac{M}{1-K} \epsilon_n \epsilon_{n+1} \epsilon_{n+2}.$$

Both the rate of convergence and the efficiency of the method, therefore, are 1.839 (see Muller [8, p. 212] for a derivation of convergence rate for a method with an error equation of the form (2.4)). In the terminology of Traub, the procedure may be classified as a one-point method with memory.

After starting, of course, we need not compute later terms of the original sequence (1.1). Instead, g is computed from ϕ as applied to successively more and more accurate estimates of the root α , and g converges rapidly to zero.

We can get an estimate for the coefficient K in the linear error term of (1.2) by forming the ratio

$$(2.5) \quad K_{n+4} = \frac{\phi_{n+3} - \phi_{n+2}}{x_{n+3} - x_{n+2}}.$$

To determine K_{n+4} in terms of ϵ_{n+2} and ϵ_{n+3} , apply (1.2) to $\phi_{n+3} - \phi_{n+2}$, thus obtaining

$$(2.6) \quad K_{n+4} = \frac{K(\epsilon_{n+3} - \epsilon_{n+2}) + L(\epsilon_{n+3}^2 - \epsilon_{n+2}^2) + \cdots}{\epsilon_{n+3} - \epsilon_{n+2}} \doteq K + L(\epsilon_{n+2} + \epsilon_{n+3}).$$

The estimate K_{n+4} may be computed and displayed each step along with the current iterate x_{n+3} .

In all three of the methods outlined, the slowly converging generating function ϕ is replaced by a function $g = x - \phi$ with a simple zero at α . It should be emphasized that with such a transformation most of the many well-known hyperlinear methods for solving $g(x) = 0$ may be utilized to find α .

3. Examples. All three of the examples are taken from [7], with computations done in quadruple precision on an IBM 370.

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

$$(3.1) \quad x_{n+1} = \phi(x_n) = x_n - \frac{f_n}{f'_n},$$

applied to a nonlinear function f with a multiple root at α of multiplicity $m \neq 1$. Tables 1 and 2 are for functions $f = (x - 1)^2 \tan(\pi x/4)$ and $f = x \sin([x - 1]^4)$, respectively. For the proposed method, the actual error $\epsilon_n = x_n - \alpha$ and the final estimate of ϵ_n from the first three error terms (2.3) are given. We can approximate the multiplicity by $m_n = 1/(1 - K_n)$ because we know K_n from (2.5) and because (see [6]) $m = 1/(1 - K)$. Both K_n and m_n are given in the tables. Results for the two methods reviewed in Section 1—the δ^2 -process and the scheme of [7]—are also included for comparison. Extrapolation substeps have a P (for prime) following the step number n in the first column.

TABLE 1

f			α	m	K	L	M	N
$(x-1)^2 \tan(\pi x/4)$			1	2	1/2	$\pi/8$	$-\pi^2/32$	$3\pi^3/128$
n	δ^2 -process ϵ_n	Ref. [7] ϵ_n	Proposed Method					
			ϵ_n	Error Estimate (2.3)	K_n	m_n		
0	-.500000	-.500000	-.500000					
1	.622531(-1)	.622531(-1)	.622531(-1)					
2	.325841(-1)							
2P	.340712(-1)	.340712(-1)	.340712(-1)					
3	.174802(-1)							
3P		-.168097(-2)	.562214(-3)					
4	.885852(-2)						.535944	2.15491
4P	-.468967(-3)	.450433(-4)	-.816331(-6)					
5	-.234397(-3)						.513264	2.05450
5P		.594677(-7)	.838173(-11)					
6	-.117177(-3)						.500220	2.00088
6P	-.863344(-7)	-.210378(-11)	.231462(-20)	.231557(-20)				

TABLE 2

f		α	m	K	L	M	N
$x \sin([x-1]^4)$		1	4	3/4	1/16	-5/64	25/256
n	δ^2 -process ϵ_n	Ref. [7] ϵ_n	Proposed Method				
			ϵ_n	Error Estimate (2.3)	K_n	m_n	
0	-.500000	-.500000	-.500000				
1	-.333043	-.333043	-.333043				
2	-.237900						
2P	-.111849	-.111849	-.111849				
3	-.829778(-1)						
3P		-.154860(-1)	.163380(-1)				
4	-.617533(-1)				.700391	3.33769	
4P	-.284075(-2)	-.494979(-3)	-.520123(-3)		.743037	3.89160	
5	-.213006(-2)						
5P		-.194741(-5)	-.415324(-6)		.750969	4.01556	
6	-.159726(-2)						
6P	-.152028(-5)	-.241102(-9)	-.111822(-11)	-.112351(-11)			

TABLE 3

ϕ		f	α	K	L	M	N
$[\exp(x-1)+1]/2$		$x-\phi(x)$	1	1/2	1/4	1/12	1/48
n	δ^2 -process ϵ_n	Ref. [7] ϵ_n	Proposed Method				
			ϵ_n	Error Estimate (2.3)	K_n		
0	-.500000	-.500000	-.500000				
1	-.196735	-.196735	-.196735				
2	-.892957(-1)						
2P	-.303500(-1)	-.303500(-1)	-.303500(-1)				
3	-.149470(-1)						
3P		-.250417(-2)	-.749119(-3)				
4	-.741794(-2)				.446848		
4P	-.218535(-3)	-.369864(-4)	-.963383(-6)		.492303		
5	-.109255(-3)						
5P		-.462123(-7)	-.408159(-11)		.499813		
6	-.546247(-4)						
6P	-.119348(-7)	-.854588(-12)	-.499176(-21)	-.499178(-21)			

Since Newton's method requires the calculation of both f and f' to get ϕ , the effective efficiency for the δ^2 -process is really only $2^{1/4} \doteq 1.189$, and for Method [7] is $(1.618)^{1/2} \doteq 1.272$. Similarly the proposed scheme with Newton's method has an effective efficiency of $(1.839)^{1/2} \doteq 1.356$.

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

$$(3.2) \quad \phi = \frac{e^{x-1} + 1}{2}.$$

We can think of the problem as that of finding a root of the nonlinear function $f = x - \phi$.

Table 3 shows results for the proposed method: the actual error ϵ_n , the final error estimate from (2.3), and the approximation K_n to K (m_n has no apparent significance in this case). Again, corresponding calculations for the δ^2 -process and for the method of [7] are listed for comparison.

Department of Energy
Argonne National Laboratory
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. N. ANDERSON & A. BJÖRCK, "A new high order method of Regula Falsi type for computing a root of an equation," *BIT*, v. 13, 1973, pp. 253–264.
3. H. ESSER, "Eine stets quadratisch konvergente Modifikation des Steffensen-Verfahrens," *Computing*, v. 14, 1975, pp. 367–369.
4. A. S. HOUSEHOLDER, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, New York, 1970.
5. R. F. KING, "A secant method for multiple roots," *BIT*, v. 17, 1977, pp. 321–328.
6. R. F. KING, "An extrapolation method of order four for linear sequences," *SIAM J. Numer. Anal.*, v. 16, 1979, pp. 719–725.
7. R. F. KING, "An efficient one-point extrapolation method for linear convergence," *Math. Comp.*, v. 35, 1980, pp. 1285–1290.
8. D. E. MÜLLER, "A method for solving algebraic equations using an automatic computer," *Math. Comp.*, v. 10, 1956, pp. 208–215.
9. A. M. OSTROWSKI, *Solution of Equations and Systems of Equations*, 2nd ed., Academic Press, New York, 1966.
10. J. F. STEFFENSEN, "Remarks on iteration," *Skandinavisk Aktuarietidskrift*, v. 16, 1933, pp. 64–72.
11. J. F. TRAUB, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, N. J., 1964.
12. 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.