SIAM REVIEW
Vol. 30, No. 4, December 1988

005

ERRATUM: Correction to "Extrapolation Methods for Vector Sequences"*

DAVID A. SMITH†, WILLIAM F. FORD‡, AND AVRAM SIDI§

Key words. topological epsilon algorithm, vector epsilon algorithm

AMS (MOS) subject classifications. 65-02, 65B05, 65F10, 65H10

In [2] we stated, with reference to the topological epsilon algorithm (TEA), "So far as we can tell, there is *no* practical implementation for TEA $\cdots$." Professor Claude Brezinski has informed us that TEA has been successfully implemented by R. Tan [3], which of course implies that our coding of this algorithm (some five years ago) must have been in error. Embarrassing as it is to have to report this error, we are pleased to have the matter straightened out, as the theoretical error analysis [1] predicts that TEA should behave much like its close cousins, the vector and scalar epsilon algorithms (VEA, SEA), also discussed in [2]. (The limiting process studied in [1] for TEA is different from the "cycling" process considered in [2], so the error analysis is not directly applicable.)

In addition to the error caused by programming bugs, there was a typographical error in the statement of the TEA formulas (5.5) and (5.6) in [2]. However, this had nothing to do with our computations, which were completed long before that part of the paper was written. For the record, the correct formulas are:

$$\boldsymbol{\varepsilon}_{2k+1}^{(n)}=\boldsymbol{\varepsilon}_{2k-1}^{(n+1)}+\mathbf{y}/(\mathbf{y}\cdot\Delta\boldsymbol{\varepsilon}_{2k}^{(n)}),\qquad k,n=0,1,2,\cdots, \tag{5.5}$$

$$\boldsymbol{\varepsilon}_{2k+2}^{(n)}=\boldsymbol{\varepsilon}_{2k}^{(n+1)}+\Delta\boldsymbol{\varepsilon}_{2k}^{(n)}/(\Delta\boldsymbol{\varepsilon}_{2k+1}^{(n)}\cdot\Delta\boldsymbol{\varepsilon}_{2k}^{(n)}),\qquad k,n=0,1,2,\cdots. \tag{5.6}$$

These formulas are for the real case only; other versions of TEA appear in [3] and in the references given there.

VEA and SEA compute triangular arrays of vectors $\boldsymbol{\varepsilon}_k^{(n)}$ in which the left boundary $\boldsymbol{\varepsilon}_0^{(n)}$ consists of terms of the sequence to be accelerated, and the right corner $\boldsymbol{\varepsilon}_{2m}^{(0)}$ (for an appropriate choice of m) is generally much closer to the limit than is $\boldsymbol{\varepsilon}_0^{(2m)}$. The computation usually proceeds along "rising diagonals" on which $n+k$ is constant, for two reasons: (a) Only the terms of the original sequence that are subsequently used need to be computed. If we compute by columns, however, we must make an arbitrary decision in advance about the length of the first column. (b) Only one diagonal needs to be retained in memory at any given time, as each computed entry may replace an entry on the previous diagonal that is no longer needed. Point (a) is significant if terms of the sequence are difficult or costly to compute; point (b) is significant if the dimension of the vector space is large.

Unlike VEA or SEA, the TEA formulas (5.5) and (5.6) for $\boldsymbol{\varepsilon}_k^{(n)}$ depend on whether k is even or odd. For the even case (5.6), the quantity being computed is on diagonal $n+2k+2$, and one of the differences requires an element on diagonal $n+2k$. Thus, the minimum number of diagonals that must be retained in memory is 1.5. Only the

* Received by the editors October 26, 1987; accepted for publication (in revised form) May 2, 1988.
† Department of Mathematics, Duke University, Durham, North Carolina 27706.
‡ Mathematical Analysis Section, National Aeronautics and Space Administration, Cleveland, Ohio 44135.
§ Computer Science Department, Technion–Israel Institute of Technology, Haifa 3200, Israel.

even-numbered entries of the diagonal two steps back are needed, and new entries can still replace old ones as they are computed.

We have now implemented both TEA and VEA in PC-MATLAB (product of The MathWorks, Inc.) and compared the two algorithms on the eight numerical examples used for other comparisons in [2], all of which are based on real scalars. The odd column entries (5.5) in TEA use an arbitrarily chosen vector $\mathbf{y}$, which we have chosen both at random and by deliberate selection. In the latter case, we selected vectors that were neither parallel nor perpendicular to either the starting vector (where that was not $\mathbf{0}$) or the known limit or antilimit. Some definite patterns emerged from these comparisons.

The following conclusions correct our statement about TEA in [2]: (a) TEA "works" as an accelerator on all eight examples, in the sense that it converges to the right answer (limit or antilimit) in all cases. Where the base sequence was convergent, TEA accelerated this convergence, but not until the second cycle in some cases. (b) In no case among these examples did the performance of TEA exceed that of VEA. (c) TEA often lagged behind VEA by two or more significant digits. (d) For some examples, the performance of TEA was almost independent of the choice of $\mathbf{y}$; for others, there were extreme and unpredictable variations with the choice of $\mathbf{y}$.

Some other details are worth mentioning. In our Examples 6 and 7 we showed that VEA could accelerate significantly even if it had the wrong information about the "essential dimension" of the problem (denoted m above), although the usual quadratic convergence deteriorated to linearity. TEA actually degraded the convergence of the base sequence in these cases, except for an occasional "lucky" choice of the auxiliary vector $\mathbf{y}$.

The PC-MATLAB computations for Example 3 in [2] lead us to another correction. The limits for the five parts of that example were computed as solutions of nearly singular systems of linear equations. We thought we had computed those limits to greater accuracy than could be achieved by the accelerators, but that was not the case. The superior "matrix divide" equation solver in PC-MATLAB has shown us that the range of accuracies achieved by VEA as the largest pair of eigenvalues goes from 0.9 to 0.99999 (previously reported as from 14 SD down to about 6 SD) should have been from machine accuracy down to 10 SD. For TEA the corresponding range is from 12.5 SD to 10 SD, i.e., it matches VEA in the most singular cases.

Now that we know how to implement TEA, our overall conclusion is similar to that for SEA: We find no detectable advantage that would lead us to prefer it to VEA in any particular case. Furthermore, optimal or even reasonable selection of the arbitrary vector $\mathbf{y}$ remains an open question, and it appears that there are situations in which this selection is crucial for achieving success with TEA.

REFERENCES

[1] A. Sidi, W. F. Ford, and D. A. Smith, *Acceleration of convergence of vector sequences*, SIAM J. Numer. Anal., 23 (1986), pp. 178–196.

[2] D. A. Smith, W. F. Ford, and A. Sidi, *Extrapolation methods for vector sequences*, SIAM Rev., 29 (1987), pp. 199–233.

[3] R. C. E. Tan, *Implementation of the topological ε-algorithm*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 839–848.