ERROR LINEARIZATION AS AN EFFECTIVE TOOL
FOR EXPERIMENTAL ANALYSIS OF THE
NUMERICAL STABILITY OF ALGORITHMS

SEppo LINNAINMAA

Technical Research Centre of Finland*, Computing Services, Revontulentie 7,
SF-02120 Espoo 12, Finland

Abstract.

After introducing briefly the principles of the error linearization method, which is able to determine the coefficients of the first order error approximation, a collection of examples is presented to demonstrate its efficiency as a test bench for analyzing numerical algorithms. These examples illustrate the propagation of initial errors, the effect of cancellation, the easy location of the most unstable parts of an algorithm, calculation of condition numbers, approximating the statistical behavior of accumulated errors and the convergence of iterative methods.


Additional Keywords and Phrases: automatic error analysis, Fortran.

1. Introduction.

Extensive work has been done to analyze the numerical behavior of algorithms mathematically. Today, several commonly used numerical methods exist where some rather simple formulas are available allowing a quite easy approximation of the effects of errors in the initial data as well as of rounding errors occurring during the computational process of the algorithm.

The amount of work involved in mathematical analysis of each algorithm is quite large, and thus some methods have been developed to automate error analysis. Probably the best known of such methods is the interval arithmetic originally developed by R. Moore [14], where each number is represented by the pair of its least and largest possible values. The disadvantage of the method is that it does not automatically take care of multiple usage of numbers and thus tends to overestimate the uncertainty of the result. E.g. if x lies between 0.4 and 0.6, then x(1 – x) is stated to lie between 0.4 · (1 – 0.6) = 0.16 and
0.6 \cdot (1 - 0.4) = 0.36 instead of the true interval (0.24, 0.25), if it is not noted that both occurrences of \( x \) obtain simultaneously the same value. Thus usually some algorithm-dependent modifications are necessary to achieve good results using interval arithmetic.

The view of the method developed by W. Miller [11, 12, 13] differs essentially from the idea of interval arithmetic. Rather than trying to approximate the interval where the result lies, a *hill-climbing* method is used to localize the worst possible initial data for an algorithm in the sense of the propagation of rounding errors. A special language was developed to represent algorithms for this analysis. A hint of the reason of instability of the algorithm can be found by locating extraordinary huge numbers in the listing representing the arguments and the result of each operation. For other approaches to automate error analysis, see e.g. [16, 4, 19, 5].

The error linearization method considered in this paper is more closely related to interval arithmetic than to Miller's method in the sense that the original data are not altered, and estimates for the interval where the result lies can be achieved. But when compared with interval arithmetic, much additional information is available allowing the user e.g. to locate fully automatically the operations which are most responsible for the instability of an algorithm. In the present article, first the error linearization method is described briefly and then a collection of examples is given presenting how this method can be utilized in various ways while analyzing numerical algorithms.

2. The error linearization method.

It is a well-known fact (see e.g. [17]) that if the numbers involved in an algorithm are denoted by \( u_i \), where \( i \) indicates the dynamic occurrence order of numbers, and if the error in the operation producing \( u_i \) is denoted by \( r_i \) and the accumulated error of \( u_i \) by \( R_n \), then \( R_n \) can usually be approximated quite accurately by the first order Taylor expansion

\[
R_n \approx \sum_{i=1}^{n} c_{n \cdot i} r_i
\]

where

\[
c_{n \cdot i} = \frac{\partial u_i}{\partial u_i}.
\]

Thus the behavior of \( R_n \) can be predicted when the partial derivatives \( c_{n \cdot i} \) and the behavior of the local errors \( r_i \) are known.

The computation of \( c_{n \cdot i} \) is not a trivial task when complicated data and control structures are used. Mutually equivalent approaches to solve this problem are presented by S. Linnainmaa [7] and M. Tienari [18], and later by J. Larson and A. Sameh [3]. The following example describes briefly the idea of this error linearization method.