SIGNIFICANCE ARITHMETIC: APPLICATION TO A PARTIAL DIFFERENTIAL EQUATION

R.L. BIVINS AND N. METROPOLIS*
University of California
Los Alamos Scientific Laboratory
Los Alamos, New Mexico 87545

Abstract-The methods of significance arithmetic are applied to the numerical solution of a nonlinear partial differential equation. Our approach permits the use of initial values having imprecision considerably greater than that of rounding error; moreover, the intermediate and final quantities are monitored so that at any stage the precision of such quantities is available. An algorithm is found that represents faithfully the solution to a difference equation approximation to Burgers' equation.

Introduction

Computational problems in the natural sciences have relatively low precision input data compared to that implied by the standard word length of a large (binary) computer. Moreover, numerical quantities are represented in the so-called normalized format with the leading digit on the extreme left of the representation of the fractional part along with corresponding associated (integer) exponent. The fractional part of a quantity may be shifted one or more places to the right with a corresponding increase in the exponent; this operation, called adjustment, leads to an equivalent, unnormalized, representation provided no meaningful bits of the fractional part are truncated on the right. This simple observation provides a degree of freedom in the representation that may be utilized to display the number of significant digits in a quantity.

Obviously the associated arithmetic processor of unnormalized quantities is not of conventional design wherein the result of the fundamental operations of addition (subtraction), multiplication or division is in normalized form. Rather, the processor must recognize the number of significant digits in each of the two input operands and produce a result with the appropriate significance. This approach is described as significance arithmetic (SA). The assumption is made that the inputs have statistically independent errors. In the course of a computation, correlations of errors can occur; such correlations must be taken into account, otherwise the actual number of significant digits in the final result may differ from the apparent number based on the unnormalized representation. That is to say, two algorithms that are mathematically equivalent may not be computationally equivalent if one algorithm properly adjusts for error correlation and the other does not.

The analysis of error propagation is a central issue in achieving a reliable algorithm. Fortunately, a technique is available that is able to localize correlations if they occur. This is the method of reduced precision [1] and is described in the sequel.

A reliable algorithm must not depend on the magnitude or precision of the input quantities that occur in problems of the natural sciences. Moreover, the interpretation of the results must be consistent with that derived from the following statistical considerations. (i) For each imprecise input, select a precise sample from an appropriate distribution, say, uniform over some appropriate interval. (ii) Execute a mathematically equivalent algorithm using normalized arithmetic; select another set of inputs from the same distributions and repeat the calculation. (iii) These repetitions lead to a distribution in each of the output quantities, whose expected value and standard deviation should agree with the approach above.

Algorithms based on significance arithmetic have been developed for linear recurrence relations, matrix inversion, least squares approximations and other problems [2], [3], [6].

In the present work we describe an algorithm for the numerical integration of a relatively simple, nonlinear partial differential equation in time and in one space variable, namely, Burgers' equation. We believe that the techniques developed are new and of general applicability. To provide the reader with some measure of the complexity involved, we also describe a much more simple sub-algorithm used in the integration scheme, specifically, the summation of a set of numerical quantities of disparate magnitudes and precisions. MANIAC II has been used; the arithmetic, as well as convenient input-output operations, are available as part of the high-level programming language, MADCAP [7].

Notation and Arithmetic Rules

1. Some notation and adjustment rules are briefly recalled [4]. Let an imprecise quantity be represented as $x = 2^e \cdot f$ where the fractional part f satisfies $-1 < f < 1$ and exponent e is an integer. It is convenient to write $x = (e,f,s)$ where s is the number of significant digits of x defined by $s = [\log_2(\bar{x}/\delta x)]_{\text{round}}$; $\bar{x}$ is the expected value of x and δx the probable error. Rounding is to the nearest integer. (Exact quantities that can be precisely represented are treated in a distinguished manner [5].)

Given two input operands $x_1 = (e_1,f_1,s_1)$ and $x_2 = (e_2,f_2,s_2)$ the result of addition (subtraction) is $x_3 = (e_3,f_3,s_3)$ where the adjustment rule is determined by

$$e_3 = \max(e_1,e_2).$$

For multiplication and division, the adjustment rule is specified by

$$s_3 = \min(s_1,s_2).$$

In practice the numerical quantity $x = (e,f,s)$ is represented in a computer by the following convention: The fractional part is adjusted so that its least significant digit resides in some bit position, say k, (counting from the left) near the right hand side of the format. The choice of k is at the option of the programmer and is consistent with the most significant quantity (apart from exact quantities) occurring in the input. Thus all fractional parts of the input are lined up on the right, rather than on the left as in normalized arithmetic. By convention, the number of significant digits in a quantity is that implied by its representation of the fractional part and consists of the digits from the k-th stage and to the left inclusively. In the sequel, it is assumed that all quantities are represented in this format.

As we shall see presently, this number of significant digits may differ from the true significance owing to the effects of correlated error. Deviations can be either positive or negative.

2. Consider a simple example of two mathematically equivalent sequences that can be computationally different. Let $x_1 = a(b - c)$ and $x_2 = ab - ac$, where a,b,c are imprecise quantities and $s_a < s_b = s_c$ and f_b is very nearly equal to f_c. It is easy to see that $s_{x_2} < s_{x_1}$ owing to the correlation of errors in *ab* and *ac*. It turns out that the number of significant digits in x_2 is actually greater than that implied by the

*This work was performed under the auspices of the U.S. Energy Research and Development Administration.

representation and that x_1 is an accurate representation of the significance.

3. The problem of summing a set of imprecise numbers requires attention to several details if a reliable estimate of the error is to be given. Let the members of the set have arbitrary magnitudes and precisions. The proper algorithm is

(i) arrange the set in a monotone non-decreasing sequence according to exponents;

(ii) add all the terms using significance arithmetic;

(iii) let n_j be the number of terms having exponent j, $j_{min} \leq j \leq j_{max}$, where j_{min}, j_{max} are the smallest and largest values respectively, and let $\sigma_j = j_{max} - j$;

(iv) form $T = \sum_{j=j_{min}}^{j_{max}} n_j 2^{-2\sigma_j}$. Finally $\tau = [\frac{1}{2} \log_2 T]$ is the number of shifts to the right imposed on the sum formed above.

The Method of Reduced Precision

If all intermediate quantities in an algorithm were free of the propagative effects of the inherent errors in the input quantities, the adjustment rules for significance arithmetic would often provide reliable estimates of the errors in the output quantities. In all but the simplest algorithms, errors are correlated in a complicated manner and one must make a series of numerical experiments to localize the point in the sequence of arithmetical instructions where correlations arise and then attempt appropriate adjustments.

The method of reduced precision (MRP) consists of

(i) make a first run with the proposed algorithm using the given (imprecise) data;

(ii) select a stage several places to the left of the k-th stage, call it k';

(iii) perturb the digits to the right of k' by means of random digits, uniformly distributed;

(iv) repeat the complete calculation with these perturbed inputs;

(v) for each output quantity, y_j, form the absolute magnitude $|y_j(k') - y_j(k)|$ defined as Δy_j and note the position of the leading digit in the fractional part of Δy_j;

(vi) repeat steps (iii)--(v) to obtain a distribution of leading digits of the Δy_j for each output quantity.

If these distributions are clustered about the position immediately to the right of k', then the original run provides a reliable estimate of the errors in the output and the algorithm and the representation is said to be faithful. On the other hand, the maximum of the distribution may be to the right, in which case more digits are meaningful than is implied by the representation of the original output; such an algorithm is conservative. Finally, the peak of the distribution may reside to the left in which case the representation of an output quantity overestimates the number of significant digits and the algorithm is liberal with respect to that output. The misrepresented digits shall be called conservative and liberal digits in the two cases.

In a complicated algorithm the results of (MRP) may not provide sufficient clues to pinpoint the sources leading to liberal or conservative algorithms. The technique is then applied in an obvious manner to intermediate quantities at various levels of bisection.

Burgers' Equation

1. We consider a difference equation that approximates the solution to Burgers' equation

$$u_t + uu_x - \epsilon u_{xx} = 0 \qquad (1)$$

where u is the velocity, the usual notation for partial derivatives is adopted, and ϵ is the diffusivity of sound. Eq. (1) is the simplest equation combining both nonlinear propagation and diffusive effects and is often used as a model for one-dimensional time-dependent Navier-Stokes equations. To first order, it approximates the motion of a plane wave of small but finite amplitude.

Our aim is to discretize Eq. (1), to numerically integrate the difference equation with initial and boundary conditions using rather imprecise data, and to specify the corresponding errors in the solution. (The procedure is sometimes called sensitivity analysis.) Because of their simpler structure, we limit the discussion to explicit difference schemes.

Several different forms were investigated; all were discarded except for the simplest, specifically,

$$u_j^{n+1} = u_j^n + \frac{\Delta t \epsilon}{\Delta x^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) - \frac{\Delta t}{2\Delta x}\bar{u}(u_{j+1}^n - u_{j-1}^n) \qquad (2)$$

where $\bar{u} = (u_{j+1}^n + u_j^n + u_{j-1}^n)/3$ and Δt, Δx are time and space intervals respectively. These intervals are determined from a classical stability analysis due to von Neumann. A variety of initial and boundary conditions have been used, the most prominent being a smooth shock wave with fixed boundary values. The number of spatial mesh points was 60. A typical set of initial values are shown in Table 1. As is well known, the solution to Burgers' equation is approximately a wave traveling to the right with velocity $\frac{1}{2}$.

$u_1^0, u_2^0, \ldots u_5^0 = 1.000$, $u_6^0 = .9500$, $u_7^0 = .5000$
$u_8^0 = .05000$, $u_9^0, u_{10}^0, \ldots u_{60}^0 = 1.0000 \times 10^{-11}$
and $u_0^n = 1.0000$, $u_{61}^n = 1.0000 \times 10^{-11}$ for all n,
$\epsilon = .010000$

Table 1. Initial values base 10 where all digits shown are treated as significant. Several sets corresponding to varying precisions were used.

The method of reduced precision (MRP) was applied to an integration over 25 cycles using 16 random samples of perturbations. MANIAC II has 44 bits in the fractional part; typically k=35 and k'=29. Using straightforward coding implied by Eq. (2) and (SA), one found the values of u in the shock region were all liberal in the sense described above, with the number of liberal digits varying from 1 to 5 and back to 1 in a bell-shaped contour across the shock. The region behind as well as considerably in front of the shock gave faithful results.

2. A useful interpretation of unnormalized representation. It is instructive to view the fractional part and the associated exponent of a numerical quantity x in (SA) in terms of information-theoretical concepts. The fractional part which carries a specification of the significant digits provides a measure of the information content. On the other hand the exponent, which has been adjusted so that the least significant digit of f_x resides in a fixed position k, gives a direct measure of the error in x. If the value u_j at some mesh point j suffers a relatively large change in one cycle, the effect appears either as a substantial change in the fractional part f or as an increase in the exponent e or both. From an information-theoretic point of view, an increase in f should not be permitted generally.

Guided by these considerations, we examined whether an increase in the number of significant digits s at any point j occurred after each cycle. If an increase did occur, the exponent was examined for any change; if not, u_j was adjusted Δs places to the right, where $\Delta s > 0$ is the observed change in one cycle. This shift is in the direction to reduce the observed liberal representation in the shock region. More specifically, if $\Delta e_j > 0$ is the change in exponent of u_j at a given point in one cycle, then we require that

$$\Delta e_j - \Delta s_j \geq 1 \qquad (3)$$

for each j. If this is violated, step-by-step adjustment to the right is made until it is eventually satisfied. The criterion specified by (3) gave faithful representations in the region of the shock. Usually $\Delta e = 0,1$ with larger values less frequently; a similar behavior was found for Δs.

3. Shock precursor. Initially, mesh points from 9 to 60 (cf. Table 1) have u-values essentially at zero. These quiescent points are excited in turn, one at each cycle beginning with point 9. Although the first non-trivial change at such mesh points is small in magnitude, about .02, the relative change is large. As a consequence of the large assumed precision initially present in the quiescent region, a certain amount of liberality gradually develops there. However by the time the shock reaches these points, the transient effect disappears. It was predicted that this initial rise in liberality would not occur if the assumed precision of the quiescent points was decreased; this behavior was verified.

4. Near equilibrium. After a shock passes through a region, the corresponding mesh points are essentially in equilibrium with each other. As they continue to be processed, their values approach each other to the extent that conservative digits appear. This effect arises because of the fact that the insignificant (and initially random) digits beyond the kth stage are handled by the computer as though they were meaningful; hence a certain convergence occurs, as might be expected under these circumstances. One easily avoids this convergence by precluding any memory of the random digits to persist.

Concluding Remarks

The present study is the first application of significance arithmetic to any differential equation. Much more remains to be done in this study of Burgers' equation. One is also encouraged that the present techniques and experience can be extended to other differential equations.

References

[1] N. Metropolis, "Algorithms in unnormalized arithmetic," Proc. Colloque International du CNRS, Besancon, France, 1966.

[2] N. Metropolis, "Algorithms in unnormalized arithmetic. I. recurrence relations," Num. Math. 7, pp. 104-112, 1965.

[3] M. Fraser and N. Metropolis, Algorithms in unnormalized arithmetic. III. matrix inversion," Num. Math. 12, pp. 416-428, 1968.

[4] N. Metropolis and R.L. Ashenhurst, "Significant digit computer arithmetic," IRE Trans. Electron. Comput., vol. EC-7, pp. 265-267, Dec. 1958.

[5] N. Metropolis, "Analyzed binary computing," IEEE Trans. on Computers, vol. C-22, pp. 573-576, June 1973.

[6] N. Metropolis and R.L. Bivins, "Linear least squares using significance arithmetic," in preparation.

[7] M.B. Wells, MADCAP Manual, Los Alamos Scientific Laboratory, Group T-7, 1964.