Robust Parallel Computation in Floating-Point and SLI Arithmetic

D. W. Lozier, Gaithersburg and P. R. Turner, Annapolis

Received March 24, 1991; revised September 3, 1991

Abstract — Zusammenfassung

Robust Parallel Computation in Floating-Point and SLI Arithmetic. In this paper we consider the parallel computation of vector norms and inner products in floating-point and a proposed new form of computer arithmetic, the symmetric level-index system. The vector norms provide an illuminating example of the contrast between the two arithmetic systems under discussion in terms of the ability to program for (complete) robustness and parallelizability. The conflict between robustness of the computation—in the sense of the dual requirements of accuracy and freedom from overflow and underflow—and easy parallelization of the algorithms within a floating-point environment is made plain. It is seen that this conflict disappears if the symmetric level-index system of arithmetic is used. The freedom from overflow and underflow offered by this system allows the programming of the straightforward definitions in a way which is simple, robust and immediately parallelizable. Numerical results are given to illustrate the fact that the symmetric level-index system yields results of comparable accuracy to those of floating-point in cases where the latter system works and still yields results of high accuracy when the floating-point system fails altogether.

AMS Subject Classifications: 65-04, 65G05

Key words: Parallel computation, vector norms, symmetric level-index arithmetic, floating-point arithmetic, error analysis, vectorized algorithms.

1. Introduction

In this paper we discuss the computation of vector $p$-norms and scalar products and the implementation of algorithms for them on vector and parallel computers. Vector $p$-norms for general $p$ are of fundamental importance, for example, in finite
element analysis [21] and multivariate interpolation [20]. The 2-norm and scalar product are central to numerical linear algebra. The 1-norm and the $\infty$-norms are important also. One purpose of this paper is to review three basic algorithms for the $p$-norm and to examine the advantages and disadvantages of each for implementation on serial, vector and parallel processors.

The calculation of $p$-norms is susceptible to overflow and underflow difficulties, especially as $p$ increases. Therefore programs, and particularly library subroutines that are intended for use on different kinds of processors, must be constructed so as to be immune to failure due to overflow and underflow. The most complicated of the algorithms that we review, an extension of Blue's algorithm [2], is developed specifically to meet this requirement. This algorithm is not a practical proposal for any architecture, but more an illustration of the lengths to which one must go for complete robustness within the floating-point system; indeed, a very simple scaling algorithm can achieve almost as much. We shall see that the extended Blue's algorithm is not well-suited to execution on vector processors, which raises the question of how the conflicting demands of complete robustness and portability between different types of processors can be reconciled.

This question leads to the main purpose of the paper, which is to suggest that floating-point arithmetic (FLP) may not be the optimal computer arithmetic for scientific computing applications. The computation of $p$-norms serves as a simple example of how FLP leads to practical difficulties in constructing robust and portable software. But the difficulties are much more widespread; they must be considered and guarded against in every seriously conceived software development project—especially where any parallel processing is involved.

Various proposals which have been made to decrease the underflow and increase the overflow limits of FLP arithmetic [13], [16] do not provide a complete solution to the problem. We propose, as an alternative, to replace FLP with the recently introduced symmetric level-index (SLI) system [6], [7], [8], [9], [15], [18], [19], [22], [23].

We introduce the definition of SLI and present some of its basic properties. The most important property for the present paper is its absolute immunity to overflow and underflow arising from any arithmetic operation (other than division by zero). This immunity is established in [15]. It holds for all finite implementations of SLI, for example in 32 or 64 bits.

Because SLI is fundamentally different from FLP, its error analysis is different from relative error analysis. The error measure, called generalized precision, is different from relative precision. Error analysis for SLI is developed in [6], [8], [17], applied to general algebraic processes in [18], and compared to relative error analysis in [15]. Only a brief introduction is included here, just enough to indicate the behavior of roundoff error in the algorithms under discussion.

In Section 2, we concentrate on the relative merits of the various approaches for computing vector norms on a serial machine and address the question of the robustness of the algorithms within a FLP arithmetic environment.