Future Directions for Numerical Software Research – Comments during Discussions at WoCo9, Prescott, AZ, July 16-21, 2006

Brian T. Smith
Numerica 21 Incorporated
Angel Fire, New Mexico, USA, carbess@swcp.com,

1 Introduction

During the discussion periods, various people expressed the concern that numerical software development had become poorly supported over the past decade, with funding agencies seeking projects that seem to ignore the development of algorithms and software for numerical computations. In the past, the community has emphasized the development of mathematical software libraries. Currently, there appears to be no interest in that mode of presentation and mode of access to our numerical software by the application community. The suggestion by Ron Boisvert and others was the Grid services may provide the new mode of access. This note discusses the relevant issues on the topic as expressed in conversation at the meeting and considered with friends since.

What is at issue here is that numerical software is hidden or buried in packages and accessed through interfaces that do not make the numerical software apparent. Because research support by the funding agencies has turned to complete application solving environments where the nature, form, and quality of the software is not directly apparent to users or those who support the development of software, support for basic numerical software has waned.

Calls for new research are for software and algorithms that work on computational grids. Approaches that address complete applications over the computational grids are paramount. In addition, there is the need for interfaces to numerical codes that are easy to use. But, libraries, with long calling sequences that provided complete control over the algorithm, do not address these needs. Many of us believe that libraries of well-tested procedures provide the building blocks for meeting the needs, but such procedures require significant expertise to use them. However, in many cases, versions of these libraries do port across the machines of most of the grid but do not necessarily provide the top performance for all machines.

The numerical software community has in the past, with a few exceptions, emphasized portability and transportability of our software. Portable algorithmic efficiency has been a goal but code tuning to particular architectures has not always been a major goal, although there have been well-known exceptions. As a result, high quality portable software is available to the application integrators from open source repositories, published papers, and commercial vendors such as NAG (Numerical Algorithms Group) and VNI (Visual Numerics Incorporated). The software from open source repositories and published papers typically is not tuned for high performance; notable exceptions are ATLAS (Automatically Tuned Linear Algebra Software) and FFTW (Fast Fourier Transformations from the West). The result is that many applications are "rolling their own software" from open sources because integrators have control over the software, including porting and performance, to some extent. The need for new basic numerical algorithms is not paramount. Only those application areas that recognize that their numerical codes are the bottle necks request or demand research in new and better software. And that recognition does not happen often.

2 Suggestion

Given the above scenario and to rejuvenate interest in numerical software, I made the specific suggestion described below that recommends a new tack be taken by our community. The suggestion assumes the interest and dominance of grid computing will continue and that this form of computing will provide lots of available cycles.

A major need in scientific applications, in my opinion, is software that evaluates the accuracy of the computed results or more importantly the sensitivity of the numerical results to the data and the platforms on which the computation occurs. (Recall on a computational grid that you may not know on what machine the computation occurred or be able to specify what machine it runs on.) Such software would run in tandem with the application code, giving assessments on the sensitivity of the results to small errors (rounding error, say) or errors in the data specified by the application. For certain numerical areas, we know how to do this; for example, matrix condition number estimators [1] can be used to indicate how sensitive a solution to a system of linear equations is to changes in the matrix elements or right hand sides, and similarly for the matrix eigenvalue problem [2]. Recently, Enright published two papers [3,4] describing techniques on how to verify the accuracy of approximate solutions for ordinary differential equations. Higham's book [5] covers many of the possible approaches, particularly for the optimization problem. The recent emphasis on noise estimation in optimization problems is likely to provide algorithms and software that address the problem of estimating the accuracy and reliability of computed solutions to optimization problems. Of course, this topic is not new; see [6] for an early introduction to the problem in estimating the reliability of computed solutions.

Software that propagates this sensitivity through the subsequent computations by the application would be required. Similar sensitivity estimates are available for ordinary differential equations, polynomial root finding and other computational processes. The numerical analysis and investigation of such techniques were studied by a number of early numerical analysis pioneers (e.g. Wilkinson, Kahan, Cody, etc.), often related to their experience with computing by hand machines. Review of their early papers may offer helpful starting points for present-day researchers.

What is needed then is research that provides the techniques for computing the sensitivity and propagating its effect through remaining computations. The approach of interval arithmetic may come to mind here but my belief is that such an approach should be used only when other more appropriate techniques are not available. The research I propose would lead to software that can run in tandem on available processors on the grid with the application, providing an assessment of the quality of the computed results. Ideally, one may be able to make such an assessment available as a grid service. It would no doubt use existing libraries of basic numerical procedures that we currently have and are developing, but needs to be presented to grid applications in an easy to use form.

References

[1] Cline, A. K., Moler, C. B., Stewart, G. W., Wilkinson, J. H., An Estimate For The Condition Number Of A Matrix, SIAM Journal Of Numerical Analysis, 16:368-375, 1979.

[2] Wilkinson, J. H, The Algebraic Eigenvalue Problem, Clarendon Press, 1965.

[3] Enright, W. H., Tools For The Verification of Approximate Solutions To Differential Equations, in Accuracy and Reliability in Scientific Computing, Edited by Bo Einarsson, SIAM, pp. 109-121, 2005

[4] Enright, W. H., JACM 195, pp.203-2006, 2006.

[5] Higham, N., Accuracy And Stability Of Numerical Algorithms, SIAM 2002.

[6] Fox, L. How To Get Meaningless Answers In Scientific Computation (and What To Do About It), IMA Bulletin, 7(1971), pp. 296-302.