Software Tools and Management for Scientific Computing

James J. Quirk

Computer and Computational Sciences Division
Los Alamos National Laboratory
Los Alamos NM87544

quirk@lanl.gov

The software content of scientific computing has grown to the point where this observation deserves debate:

> *... as programs get larger, the problems associated with their development and maintenance shift from being language problems to more global problems of tools and management.*
>
> **Bjarne Stroustrup**
> **The C++ Programming Language (2nd ed.), 1991**

This talk will explore Stroustrup's assertion from the perspective of adaptive mesh refinement (AMR):

> *...as simulations get larger, the problems associated with their development and maintenance shift from being algorithmic problems to more global problems of tools and management.*

and to avoid undue subjectivity a number of *automated case studies* will be presented that serve to illustrate the fundamental changes of perspective in moving from AMR algorithm development, through software implementation, to end application. To provide a unifying theme, particular emphasis will be given to the field of detonation instabilities.

At a practical level, this talk will demonstrate that certain *software tools and management*, packaged in an accessible form, could be used to streamline the training of the army of graduate students needed to sustain scientific computing as simulations grow ever more involved.

At a more contentious level, this talk will demonstrate the concept of a self-replicating *PDF* document and its relevance to scientific peer review. Although many journals now archive papers as *PDF* files, the format is not used to anywhere near its potential. Specifically, with bespoke *software tools and management* it is possible to embed supporting material in a journal article without increasing the printed page length. Prototype examples will be shown where AMR simulations are unpacked and run on-the-fly from a *PDF* document; for details of the programming approach see `http://www.amrita-ebook.org`.

Such dynamic *PDF* documents show the practicality of mandating self-substantiating journal articles that have the utility of hypertext documents in that they allow layered arguments and interactive simulations, and the like, but they do not have to rely on external links and so the software integrity of the end article is guaranteed. If there is a weakness of the approach, it is cultural rather than technical.

At present there is no reward system that would encourage the scientific community to want to realize a software world where all computational work could be reproduced, at the proverbial drop-of-a-hat, and so be subjected to unequivocal review. In time, however, the sheer cost of developing scientific software will likely act against such resistance and result in a community that appreciates the fundamental need to develop *software tools and management* to support co-operative working.

The immediate challenge, especially for the AMR community, is to ensure that scientific rigour is not compromised in the software-evolution process. Therefore, throughout, this talk will show practical ways to improve software accountability.