

The Benefits of Using Rigorously Tested Routines from Numerical Libraries

By Rob Meyer, Numerical Algorithms Group

Executive Summary—This white paper –“**The Benefits of Using Rigorously Tested Routines from Numerical Libraries—Composites Engineering Edition**” speaks to matters affecting nearly all corners of composites research and development. Since nearly every aspect of composites engineering and the ability to accelerate time-to-market for new materials relies heavily on computationally intensive mathematical and statistical methods, it is timely to re-examine how computational frameworks are or are not designed for maximum performance.

Although many working scientists and engineers developing new materials have long relied on pre-packaged software for many routine tasks, new areas of research have often outpaced these pre-packaged offerings. Concomitantly, whether realized or not, nearly every composites engineer today is working in a computational infrastructure that employs multicore processors that significantly SLOW performance of legacy applications originally developed for single processor computing environments.

Performance gains today, by and large, are no longer accessible through hardware upgrades—the historic path taken by commercial enterprises of all kinds through the decades. Now we have entered a period where investment in software, not hardware, may matter more. For composites engineers endeavoring to speed time-to-market, a re-examination of the computational infrastructure at work is extremely relevant.

“**The Benefits of Using Rigorously Tested Routines from Numerical Libraries—Composites Engineering Edition**” will provide a framework for understanding how commercially available numerical libraries can be a powerful tool to manage today’s composites engineering studies, including discussion of:

- commercially available numerical libraries vs. numerical code developed in-house
- commercially available numerical libraries vs. open source code
- quality control– validation and testing issues
- future-proofing– costs of migrating to newer computational platforms such as GPU or HPC environments
- future proofing—the value of library documentation vis-à-vis an application’s entire lifecycle and the given of staff turnover
- tips on real-world trials to determine ability of numerical libraries to speed application development

For more information including answers to more specific questions on the suitability of the NAG Library algorithms to particular research areas, contact David Cassell, NAG Library Specialist, at david.cassell@nag.co.uk.

How would you feel if, after devoting weeks or months to application development, you found that your application produced incorrect or unreliable results?

The use of rigorously tested routines from numerical libraries has long been a proven way to speed application development and performance. Recent developments in processor design and computational science suggests that their use should become even more widespread. This is because very few application developers have the expertise or time to assess and understand the implications of the rapidly evolving computing infrastructure on their work. As a consequence, they may not obtain the best runtime performance or consistent accuracy from their codes.

We have entered a period where investment in software, not hardware, matters most.

The advent of multicore processors is responsible for this. Traditionally performance improvements were largely attributable to the use of faster clocked processors. However in the multicore environment the clock speed of each core is now slower and hence application performance will decrease unless more than one core can be utilized. Many organizations may therefore notice degradation in application performance when they deploy the

latest hardware because they are using applications coded for use on a single processor when the new hardware is equipped with a multicore chip. Counter-intuitively, hardware that could potentially speed processing time by orders of magnitude may be responsible for significantly *slowing down* many applications. These organizations then experience significant bottom line impact without realizing why this is occurring. The issue for which they are unprepared is that programming a multicore computer is more complex than software development on a single processor system.

Numerical libraries have always been the preferred mechanism by which sophisticated technical application developers could readily incorporate mathematical and statistical functionality within their applications. Today these libraries offer a new opportunity to organizations in that they provide a convenient way to access the true power of multicore systems.

Why is this so?

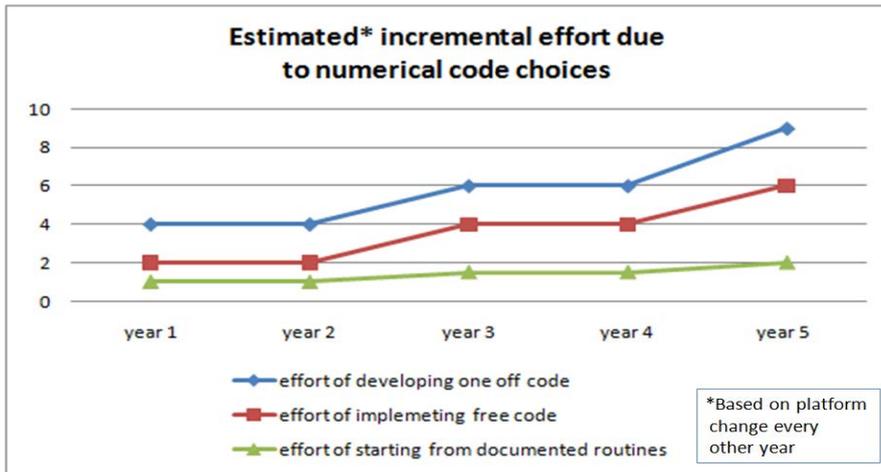
Custom developed numerical code, for use in a specific application, can be incredibly time consuming to produce and also the most costly strategy in the long term. Such code may take a long time to develop because of the complexities of designing the best-match algorithmic approach appropriate to the solution of the specific problem and the difficulty

of encoding that algorithm in an accurate and stable numerical manner. Then, the very fact that it is being written for one current application suggests that the developers may not consider the possibility of extended numerical requirements and therefore may not include the flexibility and documentation vital to enable the next advance for the product or the next development project.

It can be argued that free algorithms, available from the Internet, can provide an alternative to commercially available numerical libraries. Unfortunately support,

to choose between different hardware platforms and programming languages to best take advantage of the particular characteristics of hardware and software available but still have confidence in any results produced.

The individual numerical methods used in diverse fields such as modelling, research, analytics, design, exploration, financial engineering and product development must constantly evolve as well. This is because new, more reliable algorithms that are efficient generally or on specific hardware configurations are



constantly being developed. Indeed some routines that were considered state-of-the-art only a few years ago are thought to be very inefficient by today's standards.

Developers of numerical libraries are constantly striving, through algorithmic

maintenance and rigorous testing of these sources is at best unpredictable and therefore the user of such software is, perhaps unwittingly, risking the long term viability of their application. The risk incurred may be acceptable, in the short term, for non-critical applications but as new computing architectures emerge this risk increases significantly and may prevent optimal, or any, use of the code in the longer term.

innovations, to provide problem-solving software that is appropriate, efficient and robust on the wide range of computing platforms used at the time by the technical computing community. In this way their work continually replenishes the contents of numerical libraries with new material and improved techniques and makes these libraries available on the hardware of choice.

This latter point is important as, at any given point in time, multiple differing computing environments can be utilized by a single organization. They want to be free

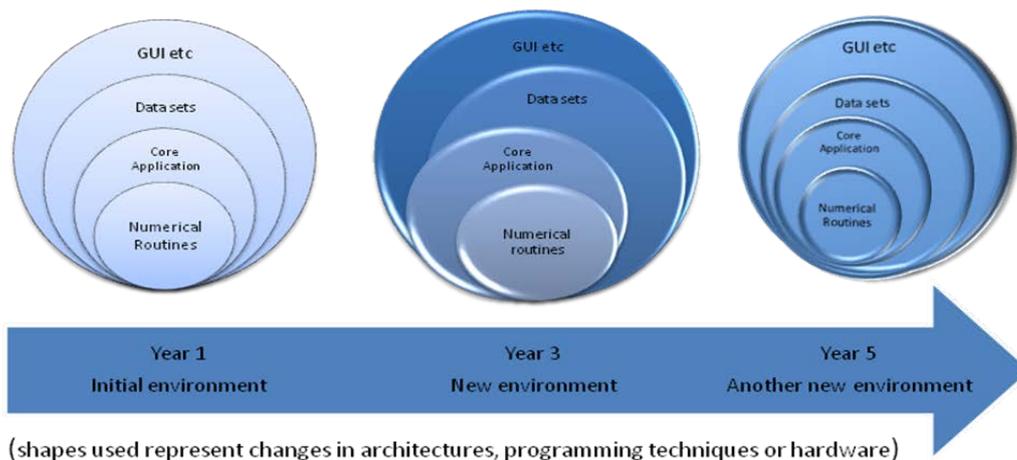
Going forward a few years another widespread shift in the normal technical computing architecture is widely predicted. This shift will have a similar range

of ramifications to the current migration from single to multicore, namely the move to manycore¹ or GPU² computing. This scenario vividly illustrates the major problem faced by organizations – investment in development of

hardware and software systems and still maintain the efficacy of their legacy code, and the many thousands of “person-hours” that it embodies, as they move forward.

Another advantage of selecting algorithms from comprehensive

The same numerical routines need to be available from different compute environments...



specialist code for a specific computing architecture may have a short lifespan before it is obsolete. In this changing environment most organizations ought not to try to justify such a cost when there are off-the-shelf alternatives produced by numerical software specialists.

The combination of the above factors put organizations that use the tried and tested algorithms from commercial numerical libraries as the building blocks of their applications, at a competitive advantage. Developers can more readily migrate codes to new

numerical libraries is that when developers require access to a different technique it may well already be available in a tried and tested form that they can rely on. This is because numerical library developers are often assisting researchers at the cutting edge, developing methods applicable to one area of research but that, over time, may well be applicable in the wider spheres of science, engineering and business analytics.

Consider as an example the inclusion, in a release of the Numerical Algorithms Group (NAG) Library, of functions for wavelet transforms. Wavelet techniques³ are used in image processing and

¹ A manycore processor is one in which the number of cores is large enough that traditional multi-processor techniques are no longer efficient. It is the generation beyond multicore.

² A GPU is a Graphical Processing Unit. General Purpose GPUs are well suited for running some numerical algorithms.

³

http://www.nag.co.uk/numeric/FL/nagdoc_f122/xhtml/C09/c09conts.xml#C09

for seismology and astrophysics. Do they come to bear on other areas of technical computing? Well yes – they are already used in the finance industry. Such cross-fertilization from one engineering or science discipline to another is a well-established pattern of how breakthroughs occur.

The extent to which numeric libraries provide extensive documentation also confers a competitive advantage to users of these libraries, especially when one considers the full lifecycle of the technical application being developed. For example, there are often numerous ways that, superficially, appear to give a developer some means to solve a

A selection of the new routines in a library release

particular problem. In practice not all of these potential approaches are effective, accurate or efficient when applied to a specific problem. The ability to select an appropriate algorithm for the task in hand is therefore vital if the developer and his organization wants timely results that can be relied upon. The detailed and refined documentation associated with well-established commercial numerical libraries

Wavelet methods in the NAG Library

The Discrete Wavelet Transform (DWT) is defined by a mother wavelet function $\psi(t)$, and a related scaling function, $\phi(t)$, where

$$\phi(t) = \sum_n a_n \sqrt{2} \phi(2t - k)$$

$$\psi(t) = \sum_n b_n \sqrt{2} \psi(2t - k)$$

These in turn are represented as a pair of filters with finite support. They can be viewed as a high pass filter, $\{h_k\}$, for $k = 1, 2, \dots, m$, paired with a low pass filter, $\{g_k\}$, for $k = 1, 2, \dots, m$. The DWT at a single level is carried out by convolution of the filter with the input data, followed by downsampling by two. In order to obtain exact reconstruction of the original input these filters must satisfy certain conditions. For orthogonal wavelets, $n = m$, these are,

$$\sum_{k=1}^m h_k = 0, \quad \sum_{k=1}^m g_k = 1, \quad \sum_{k=1}^m h_k h_{k+2l} = 0,$$

$$\sum_{k=1}^m g_k = \sqrt{2}, \quad \sum_{k=1}^m g_k^2 = 1, \quad \sum_{k=1}^m g_k g_{k+2l} = 0,$$

for all nonzero integers, l .

The reconstruction algorithm convolves the inverse filters with the wavelet coefficients previously computed together with upsampling and summation to return to the original input. For orthogonal wavelets the inverse filters are the same as those for the forward DWT.

In the simplest case, the Haar wavelet, the nonzero filter coefficients are

$$h_k = \begin{cases} 1 & k=1 \\ -1 & k=2 \end{cases}$$

$$g_k = \begin{cases} 1 & k=1 \\ 1 & k=2 \end{cases}$$

while for the Daubechies wavelet with two vanishing moments and four nonzero coefficients, the filter coefficients are

$$h_k = \begin{cases} \frac{1}{4} & k=1 \\ \frac{3\sqrt{3}}{4} & k=2 \\ \frac{3\sqrt{3}}{4} & k=3 \\ \frac{1}{4} & k=4 \end{cases}$$

$$g_k = \begin{cases} \frac{1}{4} & k=1 \\ \frac{3\sqrt{3}}{4} & k=2 \\ \frac{3\sqrt{3}}{4} & k=3 \\ \frac{1}{4} & k=4 \end{cases}$$

In the orthogonal case the same filters are used for both decomposition and reconstruction.

provide a simple mechanism by which this can be achieved. For example, the NAG Library documentation employs a decision tree structure, a tool familiar to computing professionals, to establish the precise definition of the problem to be solved. This leads the user to routines most appropriate to the actual problem.

In some areas of numerical computation, though, it is not possible to identify just one method that is most suitable for the solution of a problem. In these areas the best solution is arrived at via an iterative process whereby different techniques are applied to the problem and the best selected from the experimental evidence. Having access to a range of algorithmic techniques, using similar interfaces, to enable this experimentation can therefore be critical. The best numerical libraries provide these facilities to cater to this development paradigm.

Finally, of course, the quality of the routines provided by a commercial numerical library remains paramount. As an example, a good quality local optimization code, especially one that enables constraints on the variables, is

dauntingly long and tricky to write and can run into thousands of lines of code. If that were a custom source then the initial development constitutes only a small part of the overall cost. Other factors, such as testing, error correction and general support, which can be guaranteed when using a commercial offering, must also be taken into account.

One highly regarded library, the NAG library⁴, is continuously maintained and expanded by the Numerical Algorithms Group (NAG). The Numerical Algorithms Group (NAG, www.nag.com), is a not-for-profit numerical software development organization, with offices around the world, founded four decades ago that collaborates with world-leading researchers and practitioners in academia and industry to, among other projects, create a rigorously tested set of mathematical and statistical routines.

The routines in the NAG Library can be called from a wide range of programming languages and are relied on by those working in C++, C#, F#, FORTRAN, MATLAB, R, Maple and other environments. NAG routines are also tuned for use on multicore and parallel systems by employing standard conforming techniques in order to provide high performance while still maintaining the best algorithmic accuracy and stability. In addition, NAG's wealth of knowledge in the high performance arena is recognized by the fact that it is contracted to provide computational science and

engineering support to some of the world's leading high performance computing centres.

Researchers worldwide, collaborating with the NAG technical team, help provide regular updates and in-depth support for the NAG Library. This activity provides valuable insight into emerging requirements for new research in computer science and numerical methods. The result is one of the world's most reliable collections of numerical algorithms, including well over 1600 user-callable functions.

General purpose numerical libraries, such as the NAG Library, contain proven routines that provide access to unrivalled functionality covering a broad spectrum of numerical and statistical methods on a wide range of computing platforms. Users can readily utilize these techniques from their own programs and have complete confidence in the results returned enabling them to speedily develop sophisticated applications.

It is a straightforward task for organizations to investigate the potential benefits offered by reputable numerical libraries. For example, the mathematical and statistical library of the Numerical Algorithms Group can be used on a trial basis. Simply get in touch with NAG (http://www.nag.com/contact_us.asp) to find out which components of the NAG library will most help your work.

4

http://www.nag.co.uk/numeric/numerical_libraries.asp

Composites engineers and material scientists may also be interested in reading the following NAG case studies—

HECToR dCSE Team Quadruples Speed and Enhances Scalability of Key Materials Science Code - CASTEP

http://www.nag.com/Market/articles/CASTEP_on_HECToR1.asp

NAG Library Aids Spintronics in Magnetic Nanostructures at the University of Graz

http://www.nag.com/Market/articles/spintronics_magnetic_nanostructures.pdf

Oxford Professor uses NAG components to aid the design of new magnetic materials

<http://www.nag.com/IndustryArticles/SPECTRE.asp>

Air Products builds Complex Physical Models with the help of reliable NAG components

<http://www.nag.com/Market/articles/NLCS211.asp>

Parallel algorithms at Queen's University, Ontario - Why the NAG Library for SMP & Multicore?

http://www.nag.com/Market/articles/parallel_algorithms.pdf

NAG's HECToR dCSE Team Quadruple Performance of Quantum Monte-Carlo Application - CASINO

http://www.nag.com/Market/articles/dCSE_Report_CASINO.pdf

-30-

Rob Meyer is CEO of the worldwide Numerical Algorithms Group (www.nag.com). *The Numerical Algorithms Group (NAG, www.nag.com), is a worldwide not-for-profit numerical software development organization founded four decades ago that collaborates with world-leading researchers and practitioners in academia and industry. With offices in Manchester, Chicago, Tokyo and Taipei, and a worldwide distributor network, NAG provides high-quality computational software and high performance computing services to tens of thousands of users, from Global 500 companies, major learning academies, the world's leading supercomputing centers, numerous independent software vendors and many others.*