

---

## Applications/Algorithms Roadmapping Activity

---

### Workshop 2 Report

---

December 2008

---

## Contents

Preamble.....	2
Background .....	2
Overview of Workshop 2 .....	3
International Keynote Presentations .....	3
Osni Marques – LBNL – now at IPA assignment at the DOE HQ, OASCR.....	3
Michel Kern, INRIA and Ministry for HE and Research .....	4
Numerical Algorithmic Areas.....	5
Load balancing (meshes, particle dynamics) – Lead Peter Coveney.....	5
Optimization – Lead by Nick Gould and Jacek Gondzio.....	6
Eigenvalues: dense and sparse - Lead by Nick Higham.....	9
PDEs, domain decomposition, adaptive meshing – Lead by David Silvester .....	10
FFT and related transforms - Lead by Jan Van Lent .....	10
Iterative solvers, including Krylov methods, multigrid – Lead by Peter Jimack.....	12
Direct Methods – lead by Jennifer Scott.....	12
Software: engineering, language, libraries, validation, standards – Lead by John Brooke .....	14
HPC-NA Roadmap Development.....	15
General Considerations .....	15
Cultural .....	15
Applications and Algorithms .....	15
Software.....	16
Sustainability .....	16
Knowledge Base .....	16
Next Steps .....	17
Contacts and further information .....	17
Acknowledgements .....	17
Annex 1: Workshop Attendees .....	18
Annex 2: HPC/NA Workshop 2 Agenda .....	19

## Preamble

This report provides an overview of the second workshop of the HPC/NA Roadmapping activity. It includes all information pertaining to the workshop. The report is organized so that all outputs and likely outcomes are contained within the body of the report; supporting material is attached in annexes or may be downloaded from the project website – [www.oerc.ox.ac.uk/research/hpc-na](http://www.oerc.ox.ac.uk/research/hpc-na)

The main purpose of the HPC/NA activity is to get community input into the roadmap development. This document therefore should be seen as a report of a specific event in that activity and not as a final statement of any kind. We welcome constructive input of any sort whether to support the findings or indeed to question them.

Contributions to the discussion can happen by emailing the contacts provided below, by engaging through the project website or by attending the final workshop to be held at UCL on January 26/27<sup>th</sup> 2009.

## Background

The applications/algorithms roadmapping activity has the goal of developing the first instantiation of a high performance numerical algorithm roadmap. The roadmap will identify areas of research and development focus for the next five years including specific algorithmic areas required by applications as well as new architectural issues requiring consideration. It will provide a co-ordinated approach for a numerical algorithm and library development.

Many applications from different fields share a common numerical algorithmic base. We aim to capture the elements of this common base, to identify the status of those elements and in conjunction with the EPSRC Technology and Applications roadmapping activity, to determine areas in which the UK should invest in algorithm development.

A significant sample of applications, from a range of research areas, will be included in the roadmapping activity. The applications chosen will include those in the EPSRC Technology and Applications roadmap, and others that represent upcoming and potentially new HPC areas.

The applications should provide the basis to understand:

- The role and limits of a common algorithmic base
- How this common algorithmic base is currently delivered and how should it be delivered in the future
- What are the current requirements and limitations of the applications, and how these should be expanded
- What are the “road-blocks” that limit the scope of the future exploitation of these applications.
- A better comprehension of the “knowledge gap” between algorithmic developments and scientific deployment
- How significant computing language as well as other “practical” issues weigh in the delivery of algorithmic content

## Overview of Workshop 2

This workshop focused on the numerical and algorithmic details that had been identified in the application workshop 1. The reader is referred to the workshop 1 report and specifically to annexes 5 and 6 for the full details of algorithms and developments pertaining to that event.

There were 20 attendees including two international participants at the workshop, held at the University of Manchester – a full list of attendees is provided in annex 1. The majority were academic numerical analysts.

The agenda for the meeting is provided in annex 2. The two international participants gave overviews of activities in France and at the Department of Energy in the US [their presentations are available to download from the project website]. The rest of the programme focused on discussion around specific numerical areas to begin to understand what exists in each, what the barriers are and what planned activities there are.

## International Keynote Presentations

### Osni Marques – LBNL – now at IPA assignment at the DOE HQ, OASCR

The full presentation given by Osni may be downloaded from the project website. The slides cover a number of projects in the DOE.

Osni provided an overview of the DOE efforts in computational science applications and software. He gave an overview of the tools and libraries that are developed and supported by DOE under the Advanced Computational Software Collection: these can be found at [acts.nersc.gov](https://acts.nersc.gov). This effort started in the late 90s and continues to be supported into the future. The idea is to make the software tools available on the various computing facilities sponsored by the DOE. Education and training also feature as an important feature.

Each year DOE hold a workshop to bring together stakeholders and pay for students to attend. Osni also noted the DOE Computational Science Graduates programme that provides support for students who are able to spend time in the DOE laboratories.

The other relevant DOE program is SciDac, details of which can be found at [www.scidac.gov](https://www.scidac.gov). This supports breakthrough science enabled through HPC by partnerships among discipline experts, applied mathematicians and computer scientists.

The DOE have a number of workshops relevant to this activity planned in the next 12 months. These are under the umbrella of “High risk, high payoff technologies for applications” and are called Extreme Scale Computing Workshops in areas such as climate, high-energy physics, nuclear physics, nuclear energy, fusion, biology, and material science. Details can also be found on the website [extremecomputing.labworks.org](https://extremecomputing.labworks.org) and may provide valuable inputs to this roadmap. There are also a number of DOE reports that provide insights for this.

## Michel Kern, INRIA and Ministry for HE and Research

Michel reported on several activities that are related to the roadmapping activity. Michel's full presentation is provided may be downloaded from the project website.

Michel began with an overview of the hardware purchases in France for the provision of national HPC. France (together with the UK, Germany, Spain and the Netherlands) is one of the principal partners of the Prace project (<http://www.prace-project.eu/>) that prepares the creation of a pan-European HPC service.

Michel explained that Genci (Grand Equipement National de Calcul Intensif) provides coordination of the national centres. Genci is owned for 50 % by the French State, represented by the Ministry for Higher Education and Research, for 20 % by the CEA, 20 % by the CNRS and 10 % by the Universities. Genci also actively promotes the use of HPC in fundamental and industrial research.

He noted that while the CSCI provides the high level strategy for HPC in France, the main funding mechanism for (software related) HPC projects is the Cosinus programme from the Agence Nationale de la Recherche (<http://www.agence-nationale-recherche.fr/Intl>). These are usually 3 yr projects and generally 10-15 projects per year at a value of around 10-15 M€. The details of existing resources are given on the attached slides.

One of the items he discussed that seemed very pertinent to the roadmapping activity was the Seminar – Thinking for the Petaflop (CEA-CNRS). This project has had four working groups: Sharing, Algorithms, Organization, Teaching. These map very well onto the areas we have identified as keys for the future. The project will have a report out soon.

Michel provided an overview of several algorithmic and application software projects in France that are related to the topic of the workshop. Details can be found in the attached presentation.

## Numerical Algorithmic Areas

### Load balancing (meshes, particle dynamics) – Lead Peter Coveney

The discussion was concerned with load balancing for codes involving meshes, particle codes, and how one deals with complex interactions. Peter uses two classes of codes which are coupled with other codes. In CFD, the Lattice Boltzman algorithm is very attractive as it scales very well as we go to higher core count machines as the communications are very local. It is a nice way of modeling complex fluids and turbulence. The other class of algorithms is Molecular Dynamics (MD) with long range interactions, used for materials and biomolecular applications.

Load balancing has been a concern for many years, when trying to deploy Lattice Boltzman codes onto machines like CSAR, T3E etc. In those cases the model you are trying to describe is heterogeneous with lots of things happening in different parts of the simulations. In some applications you have interfaces in fluids across different regions or domains and you need to do a lot more computation at their interfaces. As it turned out, this class of application has never really been problematic. For example one of the codes, LB3D, has been happily studying highly heterogeneous systems and there are classes of liquid crystalline materials under flow and shear that scale well up to core counts of order 65k. Our applications are run on state of the art machines in the US such as Ranger (TACC, Austin, 62K cores) and Intrepid (The IBM Blue Gene/P at the Argonne Leadership Computing Facility (ALCF), 164K cores). The Lattice Boltzmann codes scale to those sort of numbers without too much problem probably due to the heterogeneity of the machines.

A distributed form of application can be run using MPIG (Grid extension of MPI) running over multiple machines. MPIG does a good job of overlapping communication and computation. They have examples of applications running better on two or more machines than on a single system, even if those machines are thousands of miles apart. MPIG could also be useful for running on a single system when there are heterogeneities to load balance.

In the second class, Molecular Dynamics for long range interactions, one code used is NAMDE which uses Charm++ to help with load balancing. This, in Peter's opinion, is a very messy environment for developers. Questions include how important load balancing is and how specific it is to your problem, and if there is something you can do that is generic and reusable for other people's purposes. As we go to very large core count machines, we are interested in looking at single "capability" applications that would require over half of the machine to run. Would it ever be sensible to run applications across such large number of cores? On Ranger or Intrepid, we can now run huge numbers of jobs that used to be 'large' (i.e. in the "capability" class on HPCx) and therefore do things in MD that are quite challenging such as properly sampling the system: if you can run a huge number of replicas of a system for a short period of time, the properties you get are far superior to those obtain through a single simulation.

Other people's experiences of load balancing in HPC are different. Their experience is typically problem dependent – constraints appear to be different but this might be an underlying optimization problem. Optimization libraries could be applied to load balancing problem if we were able to express the constraints appropriately and feed them into an appropriate optimization tool.

Ocean modeling and environment codes have load balancing problems of the type where each processor has to calculate according to number of sea points it has and night/day will change the

amount of radiation computation. The load balancing problems might be too specific and there might a generic approach to load balancing may not be suitable in this case.

Can these problems be offered to the Optimization community?

- Never going to get 100% perfection so solving optimization problems requires assessing the payoff
  - If it takes too long to solve to optimisation problem, then the gains are small
- This might be a max flow problem: in this case, there are good polynomial algorithms for solving this efficiently
  - There are very fast graph algorithms that can deal with this sort of problem, it's not NP hard.
- If you have access to a Grid of resources, performance and checkpointing are key, to relaunch onto more cores. The optimization issues are to do with the dynamics of the usage of the machines. MPIG is one way of handling this.

## Optimization – Lead by Nick Gould and Jacek Gondzio

For a number of reasons the first workshop did not discuss the importance of optimization. This was due in part to the range of applications included but also to the focus of the individuals involved. The DOE report *Scientific Application Requirements for Leadership Computing at the Exascale*<sup>1</sup> includes optimization as one of the key areas for the future:

*"The new algorithm categories that application scientists expect to be increasingly important in the next decade include adaptive mesh refinement, implicit nonlinear systems, data assimilation, agent-based methods, parameter continuation, and optimization."*

### APPLICATIONS

- Simulation-based optimization
  - PDE constraints
  - US DOE highlight for communication systems (civil and military) (1)
- Scheduling (electricity, gas, water)
  - network constraints
  - variables
  - global optimization
- Engineering
  - Structural design -> ideally zero-one variables

---

<sup>1</sup> Scientific Application Requirements for Leadership Computing at the Exascale (ORNL/TM-2007/238)

[http://www.nccs.gov/wp-content/media/nccs\\_reports/Exascale\\_Reqms.pdf](http://www.nccs.gov/wp-content/media/nccs_reports/Exascale_Reqms.pdf)

- Truss-topology design -> semi-definite programming
- Chemical process engineering - network nonlinear programming
- VLSI design -> semi-definite and nonlinear programming
- Contact problems -> complementarity
- Elasto-hydrodynamic lubrication -> nonlinear complementarity
- Traffic equilibrium
  - Wardrop principle -> nonlinear complementarity
- Physics
  - minimize potential energy
  - protein folding (e.g. Lennard-Jones models)
  - global optimization?
- Finance
  - Portfolio selection -> quadratic programming
  - Risk management -> linear programming
  - Asset management -> stochastic (linear) programming
  - European option/GARCH/Black-Scholes models -> nonlinear (likelihood) fitting
  - American options -> dynamic programming
  - Arbitrage models -> semi-definite programming
  - Multi-period portfolios -> robust optimization
  - (Nash) games -> complementarity
- Energy: quotes from report<sup>2</sup>
  - "First-principles computational design and optimization of catalysts will become possible at the exascale, as will novel design of biologically mediated pathways for energy conversion."
  - "Nuclear fission reactor design and optimization would help accelerate understanding of key plasma physics phenomena in fusion science"
  - "Exascale systems should also enable a major paradigm shift in the use of large-scale optimization techniques to search for near-optimal solutions to engineering problems. Many energy and industrial problems are amenable to such an approach, in which many petascale instances of the problem are run simultaneously under the control of a global optimization procedure that can focus the search on parameters that produce an optimal outcome."
  - "Of great interest are methods that will enable the power of exascale computing to advance the use of mathematical optimization in many areas of science and engineering. Examples include the use of ensembles and outer loop optimization to iterate design parameters of new nuclear reactor designs that would simultaneously improve safety margins and lower cost, or to explore the parameter space of technology choices and how they might impact global energy security strategies."
  - "Robust and reliable optimization techniques that exploit evolving architectures and are easy to use"

---

<sup>2</sup> Modeling and Simulation at the Exascale for Energy and the Environment (Town Hall meeting, 2007)

<http://www.er.doe.gov/ascr/ProgramDocuments/Docs/TownHall.pdf>

- "Appropriate algorithms for novel optimization paradigms that can be implemented only at the exascale (e.g., hierarchical optimization problems over multiple time stages) Handling of problems with hundreds of thousands of discrete parameters."
- Accelerator physics<sup>3</sup>
  - design and optimization for better efficiency at lower costs
- Meteorology
  - 4D variational data assimilation
  - O(10<sup>9</sup>) unknowns

## CURRENT OPTIMIZATION

Search (for local optima) is essentially sequential.

- Parallelism is via
  - function and derivative evaluation
  - linear system solution
- optimization often involves inequalities => needs its own (convex) analysis
- Most real optimization problems are at least NP hard!
  - non-convex optimization
  - integer programming
  - global optimization
- Optimization currently often uses implicit elimination of constraints
  - adjoints
  - inefficient optimization

NB. Often only require inaccurate solution until convergence (c.f. inner-outer iteration). Often better to use all-at-once approaches.

## OPTIMIZATION USES

- linear systems (sometimes)
- generically symmetric, usually indefinite, frequently very ill conditioned
- eigensolvers
- other solvers for constraints (ODE/PDE/quadrature)

## PARALLELISM

- branch and bound for integer and global problems

## BIG CHALLENGES

- Better polynomial methods for linear/convex quadratic programming
- Polynomial approximations to NP hard problems
- Scaling (or scale-invariant methods!)
- Derivatives (automatic differentiation)
- Good branching strategies

---

<sup>3</sup> Science Prospects and Benefits for Exascale Computing (ORNL/TM-2007/232)

- Good bounding strategies
- Warm-starting
- Semi-definite optimization (state of the art is small, systems are inevitably dense)

#### TEACHING OF OPTIMIZATION

- Needs more emphasis in the undergraduate curriculum
- c.f. Europe and North America

### Eigenvalues: dense and sparse - Lead by Nick Higham

Nick identified the standard eigenvalue transformations within an application, namely:

We begin with a rational form  $R(\lambda)x = 0$

Which generally becomes a polynomial form  $P(\lambda)x = 0$

Which is linearized to become  $Ax = \lambda Bx$

And finally  $Ax = \lambda x \quad (A := B^{-1}A)$

It is usually the latter form that application developers bring to numerical analysts to solve. Nick pointed out that we need to consider solutions over the range of forms as at each stage information is lost and often cases arise in the earlier forms.

The following table provides an overview of software available for the dense and sparse cases.

	Dense	Sparse
$Ax = \lambda x$	LAPACK	ARPACK
$Ax = \lambda Bx$		EA19 (symm) Jacobi-Davidson
$P(\lambda)x = 0$	Polyeig	
$R(\lambda)x = 0$		

It was noted that we need more detailed information regarding the sorts of applications that require eigenvalue decomposition. For example, for what problems are half or more of the eigenvectors required.

Methods for  $P(\lambda)x = 0$  are under active development and a LAPACK code for the dense case is foreseeable in the next couple of years. Very often in practice  $P(\lambda)$  has structure, such as symmetry, hyperbolicity, palindromicity or gyroscopic structure and algorithms that exploit these structures are required.

It was suggested that it would be a good idea to bring application scientists and numerical analysts together to consider the problems differently and to formulate higher level solutions. Various policy vehicles exist to enable this including from simply a workshop, to an EPSRC Ideas Factory.

This question was raised: how many problems are related to pseudo-spectra? It was noted that no one had heard an application scientist mention pseudo-spectra and it hadn't really taken off as a standard tool at this stage.

## **PDEs, domain decomposition, adaptive meshing – Lead by David Silvester**

The last workshop identified adaptive mesh refinement as a key area with six of the presentations indicating that this was an area of concern for them. It was noted that the UK has some disparate groups working in this area including groups at Bath, Imperial, Leeds and Manchester and some in Oxford.

David identified the PDE's that are being solved using HPC:

- Navier Stokes
- RANS
- Schrödinger Equations
- Porous differential equations
- No mention of elasticity
- Maxwells equations
- Einstein Equations
- Stochastic PDEs

Codes are running faster with bigger machines however the algorithmic approach will need to change with parallelism and multi-core. Many people are still using packages developed in the 1970s. The new architectures won't be suited to these old codes. The codes are increasing in complexity as more physics is being added to the model, higher resolutions are being used and there are more coupled models.

## **FFT and related transforms - Lead by Jan Van Lent**

(with input from Ivan Graham & Rob Scheichl)

FFTW<sup>4</sup> is the standard high performance software for FFT which is widely used. From wikipedia: FFTW, for "Fastest Fourier Transform in the West," is a software library for computing discrete Fourier transforms (DFTs) developed by Matteo Frigo and Steven G. Johnson at the Massachusetts Institute of Technology. It is public domain under GNU; there is also a commercial version from MIT and it underlies

---

<sup>4</sup> [www.fftw.org](http://www.fftw.org)

the `fft` and `ifft` commands in MATLAB. FFTW handles data of any length  $N$ , but works best when  $N$  has small prime factors: powers of two are optimal size; a (large) prime sizes provide the worst cases.

FFTW is widely used by the scientific community in particular computational Physicists and Chemists.

FFTW runs under MPI so supports parallelism but its parallel performance is controversial. Several participants at the Oxford Roadmapping meeting reported dissatisfaction. Recent versions of FFTW are optimised for special architectures like multicore, cell, GPU or FPGA's.

There is a page on parallel FFT at <http://www.sandia.gov/~sjplimp/docs/fft/README.html>

FFTW assumes equally spaced data but there are recent versions of FFT for non-equally spaced data.

The NFFT (a form of the FFT that allows non-equally spaced sample) has been developed by Daniel Potts (now at Chemnitz) and co-workers: [www-user.tu-chemnitz.de/~potts/nfft](http://www-user.tu-chemnitz.de/~potts/nfft). A number of FFT spin-offs are offered at their web site: for example, a fast Gauss transform, a fast summation of radial functions on the sphere, polar FFT, etc. Key names are Keiner, Kunis and Potts.

In the USA, a more popular (and older) version of the FFT that allows non-equally spaced data is called USFFT (Unequally spaced FFT), see [www.fmah.com/IMAGES/SEISMIC/MANUAL.PS](http://www.fmah.com/IMAGES/SEISMIC/MANUAL.PS)

#### OTHER TRANSFORMS

A, by now, fairly old fast Legendre transform software written by Mohlenkamp is available. This involves computational costs proportional to  $C N \log N$ , unfortunately with a large constant  $C$ , so that it can only be used efficiently for large values of  $N$ . As far as we know, this has not been widely successful.

A fast Legendre transform together with an FFT could be used for fast computation of spherical harmonic expansions for functions on spherical domains such as occur in many problems in geophysics. Important names in this area are Driscoll and Healy and more recently Kunis and Keiner, Suda and Takami. The big group of Freeden in Kaiserslautern makes heavy use of such technology in geophysical applications. There are at least two suites of software to do "spherical FFT": [www.cs.dartmouth.edu/~geelong/sphere/](http://www.cs.dartmouth.edu/~geelong/sphere/) and [www-user.tu-chemnitz.de/~potts/](http://www-user.tu-chemnitz.de/~potts/)

It was agreed that there is international demand for such transforms e.g. in fields like Numerical Weather Forecasting. One example is Theora available through the Xiph.org Foundation: [www.theora.org/faq/#32](http://www.theora.org/faq/#32)

The JPEG2000 standard uses wavelet compression: [en.wikipedia.org/wiki/JPEG\\_2000](http://en.wikipedia.org/wiki/JPEG_2000)

Many examples of wavelet compression methods for pictures and images are listed at [en.wikipedia.org/wiki/Wavelet\\_compression](http://en.wikipedia.org/wiki/Wavelet_compression)

Many technology companies, cameras etc, are working on video compression and the like (including the BBC).

## **Iterative solvers, including Krylov methods, multigrid – Lead by Peter Jimack**

Peter noted that while the implementation of most iterative methods in parallel is not particularly demanding, the development of effective, parallel preconditioners is very challenging. Peter focused on sparse problems although he noted that sparse preconditioners may also be employed with boundary element methods.

Peter reported how, for most cases, preconditioners based on the approximate inverse could be used, and gave an overview of methods for calculating approximate inverse preconditioners, highlighting their strengths and weaknesses with regards both to the problem domains and to the type of hardware architecture. It was noted that often the structure of the matrix and physical properties of the problem can provide clues.

A number of packages were discussed including HYPRE from Livermore and AZTEC from Sandia National Labs. It was clear that there is much more activity in the US than the UK in this area.

Nick Gould stressed that reliable, stopping criteria for convergence need be developed. He also asked at what level parallelism would be most effective for non-linear equations: in the inner iterations (i.e. the underlying linear iterative solvers) or in the outer iterations (i.e. exploring in parallel the non-linear equation solution space).

Peter Jimack noted that there are applications that need to provide reproducible results: this imposes considerable constraints on any algorithms using an asynchronous or random communication pattern.

### **COMMENTS ON THE “ITERATIVE SOLVERS” DISCUSSION**

- Many systems are structured, either globally or through structured sub-matrices -> structure-exploiting not generic methods needed
- A key issue is when to stop (often determined by physics, etc)
- All issues discussed also relevant for nonlinear equations, but then a delicate balance between how accurately to solve inner (linear) system vs. pay-off for overall progress
  - trade time for function/derivative evaluations against time for linear solvers

## **Direct Methods – lead by Jennifer Scott**

Basically, there are two separate cases: dense and sparse. The dense case is the remit of such projects as BLAS, LAPACK and ScaLAPACK. The UK has had involvement in this over the years (notably, people at NAG, RAL and Manchester). Much of the current effort is being done in the US, in particular, Jack Dongarra and his team is leading the way with the PLASMA project.

Dense	Sparse
BLAS	MUMPS -MPI
LAPACK	Pardiso – Basel, Intel, OpenMP
SCALAPACK	WSMP - IBM
	SuperLU
	PASTIX

Most sparse direct solvers take advantage of software for dense solvers, as they rely for efficiency on dense. However, improved dense kernels only lead to modest improvements in sparse solvers: it is necessary to design new sparse algorithms that can exploit more general parallelism. This is a tough but very important problem because the solution of sparse linear systems often lies at the heart of computational science, engineering and finance problems.

As models become more sophisticated, ever larger systems need to be solved accurately and efficiently. A small number of parallel sparse solvers are available. Main codes are:

- MUMPS: developed in France. This is an MPI-based code.
- PARDISO: originally developed at the University of Basel. It has now been taken over by Intel and is distributed with the latest version of the Intel Library. It uses OpenMP.
- SuperLU: it comes from Berkeley and is widely used in the USA. It was designed for unsymmetric systems only. There are versions for both shared and distributed memory.
- WSMP: IBM code (commercially available).

All of these codes have drawbacks and none will solve all the problems users are currently interested in.

Memory is a key issue for direct solvers. Recently, there has been considerable interest in working out of core, but this is another challenge in the parallel case.

Jennifer noted that we should stop looking at direct methods as black boxes but look more carefully at the structure of a problem. We should probably also research into hybrid methods, crossing over the separation between direct and iterative methods as, for example, iterative refinement techniques become more important.

#### COMMENTS ON THE "DIRECT SOLVERS" DISCUSSION

- iterative refinement should be employed - blurs distinction between iterative and direct methods
- sometimes require many solves per factorization - pays off to compute "sparse-est" possible factors
- third class of problems which are dense but structured - loose information if simply use LAPACK
- under/over-determined systems very important - least squares and regularisation methods needed (QR vs. LU)

## Software: engineering, language, libraries, validation, standards – Lead by John Brooke

John began by saying how a well documented body of research showed that usability (how software is viewed and approached by users) was an often contentious issues issue, arising, for example, in the choice of programming languages, development methodologies, look-feel of a package, etc. This can have an even greater impact than algorithmic choice. These questions were seen as paramount:

- Who may use the software, the actual products of algorithms?
- What are the groups of users of these algorithms and what is their degree of sophistication. in the context of HPC?
- How do they want algorithmic content to be delivered: libraries, own implementation of algorithms, components of general packages like Matlab?
- Why do they need algorithms and software? The needs of scientific applications, aimed at research and discovery, or engineering where validation and verification actual physical model are essential, may well differ.
- What machines do they want to use? Hardware architectures affect algorithms, of course.

If we can answer these questions to some extent then they will help with thinking about the ‘engineering’ aspects of the numerical algorithms. In some cases the results of the algorithms need to be reproducible and repeatable, and in others the ability to do discovery are more important. In general the whole body of practice in software engineering maybe hasn’t been taken up sufficiently in the science domain.

In terms of reliability we can’t ignore underlying architectural issues. Very large systems by their very nature are likely to have physical errors of nodes or memory – as their size increases so does the probability of hardware/software failure. Parity errors are a particular problem. Systems are designed to continue even if a node fails.

The final point is about distribution. How are the software products going to be used? Will they go into libraries, or to commercial partners for hardening, or is the aim to help the actual users of the algorithms to incorporate them into their own user-generated codes? There will be different answers for different users, which has an impact on how the software is engineered.

### Discussion

- It is important to bear in mind that the best delivery system may not be a library – could still be appropriate for a numerical service – particularly, for example, on a GPU. Integrating physics into the model might be easier if not calling library codes
- “Frameworks” may become essential for delivery.
- Some codes are “horrendous” because of complexity and because of poor documentation

Matthias Heil noted that we need to make sure that the framework doesn’t get so big that we lose the capability to deliver it.

There was a lot of discussion of frameworks and object oriented approaches that have been attempted in the past such as CCA and PETSc. It is clear that some have been more successful than others and we need to learn from those experiences.

## **HPC-NA Roadmap Development**

### **General Considerations**

The reader is referred to the workshop 1 report for the thinking on the roadmap. The draft roadmap is now developed in a separate document – Algorithms and Applications Roadmap draft 1.0. The workshop participants reviewed the Utopia as defined by the first workshop. There were a number of comments that led to a slight change of the identified goal:

The Grand Challenge is to provide

- Algorithms and software that application developers can reuse in the form of high-quality, high-performance, sustained software components, libraries and modules
- a community environment that allows the sharing of software, communication of interdisciplinary knowledge, and the development of appropriate skills.

It was noted that we should also focus on understanding and considering the UK areas of strength to make sure that investment and development build on them. The International Review of Mathematics (2004) highlighted linear algebra, multiscale and adaptive algorithms, stochastic differential equations, preconditioning techniques and optimization as areas of strength in numerical analysis and scientific computing in the UK.

The areas of the roadmap were expanded upon as follows:

### **Cultural**

- a. Identify potential community players
- b. Develop models of community sharing
- c. Provide community activities, workshops, training, virtual meeting spaces.
- d. Engage internationally

### **Applications and Algorithms**

- a. Identify exemplar applications
  - i. Develop baseline models for communication and benchmarking
- b. Develop map of algorithms across application domain
  - i. Identify impact of specific algorithm development across discipline groups
  - ii. Speed dating
  - iii. Take mapping of dwarfs on capability computing
- c. Develop map of developments internationally
  - i. Collect information about ongoing related activities
  - ii. Discuss with international funding agencies plans

## Software

- a. Abstractions (in collaboration with CS)
- b. Code generation and adaptive software systems
- c. Guidance on best practice for software engineering development
- d. Develop frameworks and tools for application developers
- e. Languages = take note of the DOE funded activities.

## Sustainability

- a. Develop models for sustainable software
  - i. Long term funding
  - ii. Industrial translation
  - iii. Open community support
  - iv. Other
- b. Creation of MSC and other postgraduate training

## Knowledge Base

- a. Develop mechanisms for collecting information on existing software and dissemination
- b. Develop mechanism for continuing community input
- c. Education and training –
  - i. Optimization for example
  - ii. Software engineering
  - iii. Provide computational science internships
  - iv. Bid for short courses or summer schools

It was suggested that we need to have a repository of expertise of NA in the UK and we should consider mechanisms for matching people in appropriate teams.

It was noted that the DOE have a similar exercise under way and have asked Jack Dongarra to organize workshops to discuss these issues, one in US, one in Europe, and one in Asia.

## Next Steps

The roadmap is being further developed from the outputs of this workshop and is now in a separate document.

We are continuing to collect further input from applications scientists and would welcome input from numerical analysts on this report and the activity as a whole as well as further input from computer scientists.

The next workshop will take place on January 26/27 at the Royal Society, London, where the roadmap developed will be presented to all stakeholders for review.

## Contacts and further information

### Issues and input to this Report

Dr Mark Hylton:

[mark.hylton@oerc.ox.ac.uk](mailto:mark.hylton@oerc.ox.ac.uk)

### General input to Activity

Prof. A. E. Trefethen, OeRC, University of Oxford

[anne.trefethen@oerc.ox.ac.uk](mailto:anne.trefethen@oerc.ox.ac.uk)

Prof P. V. Coveney, University College London

[p.v.coveney@ucl.ac.uk](mailto:p.v.coveney@ucl.ac.uk)

Prof N. J. Higham, University of Manchester

[Nicholas.J.Higham@manchester.ac.uk](mailto:Nicholas.J.Higham@manchester.ac.uk)

Prof I. S. Duff, STFC, Rutherford-Appleton Laboratory

[iain.duff@stfc.ac.uk](mailto:iain.duff@stfc.ac.uk)

### Project website

[www.oerc.ox.ac.uk/research/hpc-na](http://www.oerc.ox.ac.uk/research/hpc-na)

## Acknowledgements

We are grateful for the support provided by EPSRC for the algorithm/application roadmapping activity. Thanks also to Nick Higham and colleagues at Manchester for organizing and hosting the second workshop.

## Annex 1: Workshop Attendees

<b>Name</b>	<b>Institution</b>
Michel Kern	INRIA, France
Osni Marques	LBNL & DOE, USA
Nick Higham	University of Manchester
David Silvester	University of Manchester
Jan Van lent	University of Bath
Jacek Gondzio	University of Edinburgh
Peter Jimack	Leeds University
Iain Duff	RAL, STFC
Nick Gould	RAL, STFC
Anne Trefethen	University of Oxford
Mark Hylton	University of Oxford
Stefano Salvini	University of Oxford
Peter Coveney	UCL
Stephen Pickles	Daresbury, STFC
Emma Jones	EPSRC
John Brooke	University of Manchester
Ben Leimkuhler	University of Edinburgh
Jennifer Scott	RAL, STFC
Milan Mihajlovic	University of Manchester
Matthias Heil	University of Manchester

## Annex 2: HPC/NA Workshop 2 Agenda

Manchester Institute for Mathematical Sciences (MIMS), Alan Turing Building, Frank Adams Room 1

### Day 1: Monday 8th December

13:00-14:00 Lunch

First Session Chaired by Nick Higham

14:00-14:30 **Presentation of summary of results from previous workshop.**

14:30-15:00 **Load balancing (meshes, particle dynamics)**

Lead: Peter Coveney

15:00-15:45 **Osni Marques**, Lawrence Berkeley National Laboratory

15:45-16:00 Refreshments

Second Session Chaired by Iain Duff

16:00-16:45 **Michel Kern**, Inria

16:45-17:30 **Optimization**

Leads: Jacek Gondzio and Nick Gould

17:30-18:00 **Eigenvalues: dense and sparse.**

Lead: Nick Higham

18:45 Dinner at Tai Pan Restaurant, Upper Brook Street

### Day 2: Tuesday 9th December

09:00-09:45 **PDEs, including domain decomposition, adaptive meshing.**

Lead: David Silvester

09:45-10:00 **FFT and fast transforms.**

Lead: Jan Van lent

10:00-10:30 **Iterative solvers, including Krylov methods, multigrid.**

Lead: Peter Jimack

10:30-11:00 **Direct solvers.**

Lead: Jennifer Scott

11:00-11:15 Refreshments

11:15-11:45 **Software: engineering, language, libraries, validation, standards.**

Lead: John Brooke

11:45-12:45 **Initial development of roadmap**

12:45-13:45 Lunch

13:45-15:00 **Further development of roadmap and plans for third workshop**