Developing a high performance computing / numerical analysis roadmap

Prof. A. E. Trefethen, University of Oxford
Prof. N. J. Higham, University of Manchester
Prof. I. S. Duff, Rutherford Appleton Laboratory
Prof. P. V. Coveney, University College London

Applications/Algorithms Roadmapping Activity

Roadmap Version 1.0

January 2009

EPSRC

http://www.oerc.ox.ac.uk/research/hpc-na

Contents

Preamble

This first version of the HPC/NA roadmap should be considered as a live document. The main purpose of the HPC/NA activity is to get community input into the roadmap development and we recognize that this first version has been developed with input from only a selection of the relevant community. As further input is gained we expect that it will naturally impact and change the roadmap presented here.

The HPC/NA roadmap activity is funded by the EPSRC. The outcomes of the activity should inform the research programs within EPSRC.

Executive Summary

This first version of a roadmap document is the outcome of two community meetings together with input from similar activities elsewhere. The roadmap activity aims to provide a number of recommendations that together will drive the agenda toward the provision of

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

This first version of the roadmap is built around five themes that have evolved during the discussion within the community. Each of these is represented in the roadmap and here we suggest some initial actions for further consideration. As the roadmap activity goes forward we expect that these initial actions to develop into a detailed map of priorities across a sensible timeframe.

Theme 1: Cultural Issues

There is a need to

a. Identify potential community players across application domains, numerical analysis and computer science
b. Develop models of community sharing of algorithms, software and ideas
c. Provide community activities, workshops, training, virtual meeting spaces.
d. Engage internationally

Theme 2: Applications and Algorithms

There is a need to

a. Identify exemplar application to develop baseline models for communication and benchmarking
b. Develop a map of algorithms across application domain
    i. Indentify impact of specific algorithm development across discipline groups
    ii. Take mapping of dwarfs or similar on capability computing
c. Develop map of developments internationally
    i. Collect information about ongoing related activities
    ii. Discuss with international funding agencies what plans are in place in this area

Theme 3: Software

There is a need for

a. Abstractions (in collaboration with CS) to allow more effective application development
b. Code generation and adaptive software systems to automatically deliver efficient code for complex architectures
c. Guidance on best practice for software engineering development
d. Frameworks and tools for application developers to allow better reuse of algorithms
e. Better understanding of usability issues for complex software systems

Theme 4: Sustainability

In this theme the issue of sustainability applies to both software and skills, but we take up the latter under the final theme. Hence the requirement here is to develop models for sustainable software through vehicles such as long-term funding, industrial translation and open community support.

Theme 5: Knowledge Base

This theme is concerned with the general issue of sharing of knowledge and knowledge creation. The recommended actions are:

a. Develop mechanisms for collecting information on existing software and expertise and dissemination
b. Develop mechanism for continuing community input
c. Develop appropriate education and training, through MScs, DTCs, short courses and summer schools. Engage industry, possibly through internships, to ensure industry needs are also met.

The Challenge

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 application development.

Many applications from different fields share a common numerical algorithmic base. The roadmap aims 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 has been included in the roadmapping activity and we are interested in adding any new or underrepresented area.

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 the current requirements and limitations of the applications are, 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 languages as well as other "practical" issues weigh in the delivery of algorithmic content

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 that lead to a better capability to develop high-performance applications.
- A community environment that allows sharing of software, communication of interdisciplinary knowledge, and the development of appropriate skills.

We hope that the roadmapping activity will also help to identify UK strengths and weaknesses to assist in choosing appropriate investment areas.

Background

The Changing Face of HPC

High performance computing (HPC) has moved from being a somewhat esoteric interest of a few "bleeding-edge" scientists to a necessity for any computational scientist, any software developer, and any industry that uses simulation as a tool. This shift has come about as processor chip designs now mimic the architecture of high performance computers, with multiple processing cores on a single chip, making efficient programming of a single processor computer as complex as it once was to develop software for a high performance computer. Furthermore today's HPC systems comprise many thousands of such processors connected by a high performance switch requiring hierarchies of different programming models.

Advanced computing is an essential tool in addressing scientific problems of national interest, including climate change, nanoscience, the virtual human, new materials, and next-generation power sources, but as importantly it is equally essential to solve commercial and industrial problems in financial modelling, engineering, and real-time decision systems. Complex surgery, tumour imaging and cancer treatment, effects of drugs on the human system, and many more health-related applications increasingly depend upon advanced computing. It is notable that of 53 UK computer systems included in the top 500 supercomputers (www.top500.org) 15 are listed as financial and a further 13 are companies providing services to the finance sector. A standard dual-core laptop is equivalent to one of the top500 machines of only 12 years ago. High performance computing is no longer for the select elite!

Yet our capability to use these machines is diminishing with the increasing complexity of hardware. The algorithms underlying current software are not able to cope efficiently and are not able to fully exploit advanced computing architectures. We can no longer take it for granted that for each successive generation of microprocessors the software applications will immediately, or after minor adjustments, run substantially faster. Companies who supply computational software for simulation development will be hit by the fact that all underlying algorithms will need to change to take advantage of the new evolving machine architectures or will likely run slower on newer computer systems.

As a nation we are poorly equipped to address these challenges. There is a lack of cohesion across the disciplinary groups that need to be brought together, namely mathematics, computer science, engineering and domain scientists. There are few existing channels to allow a flow of knowledge and expertise across the academia/industry divide. We have an inadequately trained next generation of researchers and we do not have the required skilled workforce.

There have been a number of studies in the USA [1, 2, 3] focused on this issue and we aim to leverage their findings. The recent report on exascale computing for energy and environment notes "The current belief is that the broad market is not likely to be able to adopt multicore systems at the 1000-processor level without a substantial revolution in software and programming techniques for the hundreds of thousands of programmers who work in industry and do not yet have adequate parallel programming skills." [4]

It is widely recognized that, to date, algorithms and libraries have contributed as much to increases in computational simulation capability as have improvements in hardware [7]. The developments in computer systems throw even greater focus on algorithms as a means of increasing our computational capability[2]. Enhancing the national capabilities in advanced computing algorithms and software will have a major impact on the UK's future research capacity and international impact in the ever increasing number of domains within which high performance computing is, or is set to become, a core activity. In addition to all the domains within EPSRC's own remit, there are many areas of interest to other UK Research Councils for which it is absolutely essential that the UK make strategic investments now in high performance computing algorithms and software to safeguard our ability to be internationally leading in the next ten to twenty years and beyond.

In the UK, the OST Science and Innovation Framework 2004-2014 [8] identified the need for computational research and the underpinning research infrastructure to support it. The International Reviews of Mathematics (2004) [11] and High Performance Computing (2005) [12] both identified the UK as having internationally leading research groups in the areas of numerical analysis and HPC; however, the lack of collaboration between the numerical analysis community and HPC researchers was also identified by both as a significant concern that will result in the UK missing important scientific opportunities. In countries such as France, Germany and the USA that have large government funded laboratories there are strong links between heavy users of numerical algorithms for solving grand challenge problems and numerical analysts and computer scientists in academia, with significant funding from various agencies to facilitate these collaborations (for example see the US DOE SciDAC effort [5]). In the UK, such links between scientists in academia and industry are underdeveloped and do not lead to the symbiosis that we see in these other countries. The consequences of the UK situation are that algorithmic breakthroughs are slow to be picked up and used by computational scientists, and numerical analysis research efforts are not necessarily being targeted at the problems of most interest to the scientists.

The roadmap aims to bring these communities together, to collaboratively identify the algorithmic and computational areas that need investment to ensure the UK is able to compete internationally. It will identify potential priority areas where the UK can have impact, and will propose actions to bring together the stakeholders to take those priority areas forward.

Architectural Trends

Physical barriers mean that while the number of transistors per chip continues to double at the historic rate, processor clock rates are ceasing to increase. This is the primary driver for chip designs changing to multi-core architectures. Intel will ship 6-core this year, 8-core next year, and the 16-24 core Larrabee chip in 2010. Meanwhile, NVIDIA is already providing 128-core today to 256-core later this year, and AMD's forthcoming GPU-based product is 320 core. But the number and throughput of pins on chips are reaching their limits, meaning that multi-core chips will have an increasingly large gap between processor performance and memory performance.

Meanwhile, economic necessity forces the use of many thousands of commodity CPUs to scale up to the next generations of high end systems. These factors mean that the HPC community can no longer take it for granted that each successive generation of microprocessors will immediately, or after minor adjustments, result in their software running substantially faster. Hence in order for scientists to tackle increasingly complex problems using HPC, algorithms will need to be developed that employ novel mathematical and coding techniques. Moreover, new and revised algorithms will need to be rapidly translated into software developed with careful adherence to standards and portability---which in turn support both maintainability and adaptability---to ensure a long lifetime.

Relevant activities

There are a number of activities in other countries that are relevant to the roadmap and where possible we would like to leverage and borrow from those strategies. It is important that any roadmap for UK activity makes sense within the context of the global picture.

In the second workshop we learned from Michel Kern of INRIA about the “Thinking for the Petaflop” activities in France where the initial areas of interest coincide with this roadmap. At the time of writing the report from these activities has not yet been published but we expect to feed those results into this roadmap.

There have been a number of significant activities in the USA and perhaps the three most notable such that are highly relevant to this activity are the ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems [3], the DOE Modelling and Simulation at the Exascale for Energy and the Environment [4] study and the Oak Ridge report on Scientific Application Requirements for Leadership Computing at the Exascale [6]. The first of these reports provides a comprehensive review of both hardware and software issues for exaflop computing, together with the practical issues surrounding development of suitable data centres for exaflop computers.

The reports have focussed on key application areas such as energy, climate, weather simulation, biology and astrophysics, which not surprisingly mirror many of the applications that are of interest here. However all these reports are focussed on exascale computing, which is the peak of high-performance computing. Our interests are broader than that and we are hoping to enable a wider range of capability computing applications.

Algorithms and Application Roadmap

Methodology

A full description of the methodology used to develop the roadmap is provided in Annex 1. In essence the effort has been a combination of background desk work, a series of workshops and a collaborative community site. The latter has not provided input to version 1.0 of the roadmap but should provide significant input in the future. The three workshops held in Oxford, Manchester and London are described in full in Annex 1; they brought together applications developers, numerical analysts, computer scientists, industry scientists and computer vendors. The outputs from the workshops have been distilled and circulated to the broader community.

Application Domains and Issues

Within the activity to date we have focussed on applications that are of interest or importance within the UK. There has been an attempt to identify new and upcoming application areas as well as those that are long established. A major issue that all the application areas face is scaling. There are generally four reasons that applications want to be able to scale:

1. A desire to include more realistic physical models. This implies higher resolution, more physical parameters and in general a greater complexity.
2. Simply to solve a larger problem with the same physical parameters.
3. A need to move to real-time simulation; to be able to solve the same problem but much faster.
4. A desire to complete far more time steps of the same simulation.

A good analysis of the requirements on a computer architecture, with respect to memory, storage and communications, is given in [6] for each of these categories. There it is also noted that the maturity of algorithmic approaches within applications, such as adaptive mesh refinement, mean a complexity in the application that did not exist with regular grids and the like. Similarly the complexity of the hardware with multi-core processors and vector units combined in 10s of thousands of processor units means that algorithms need to be designed to be efficient across a broad range of memory hierarchies and chip architectures.

Both [6] and [3] provide mappings of application areas to algorithmic requirements. Figure 1 shows the table from [6] which was developed through an analysis of the applications at Oak Ridge and using the seven dwarf classification of Colella - seven numerical methods that he believed will be important for science and engineering for at least the next decade [9].

Table 2. Algorithms expected to play a key role within select scientific applications at the exascale, characterized according to a seven dwarfs classification

Opportunity	Application area	Structured grids	Unstructured grids	FFT	Dense linear algebra	Sparse linear algebra	Particles	Monte Carlo
Material science	Molecular physics			X	X		X	X
	Nanoscale science	X			X		X	X
Earth science	Climate	X	X	X		X	X	X
	Environment	X	X			X	X	X
Energy assurance	Combustion	X			X		X	
	Fusion	X	X	X	X	X	X	X
	Nuclear energy		X		X	X		
Fundamental science	Astrophysics	X	X		X	X	X	
	Nuclear physics				X			
	Accelerator physics		X			X		
	QCD	X						X
Engineering design	Aerodynamics	X	X		X	X		

Figure 1: Application analysis from [6], Scientific Application Requirements for Leadership Computing at the Exascale

Figure 2 shows a similar analysis provided in [3] with its origin in the work of David Keoster & others [10].

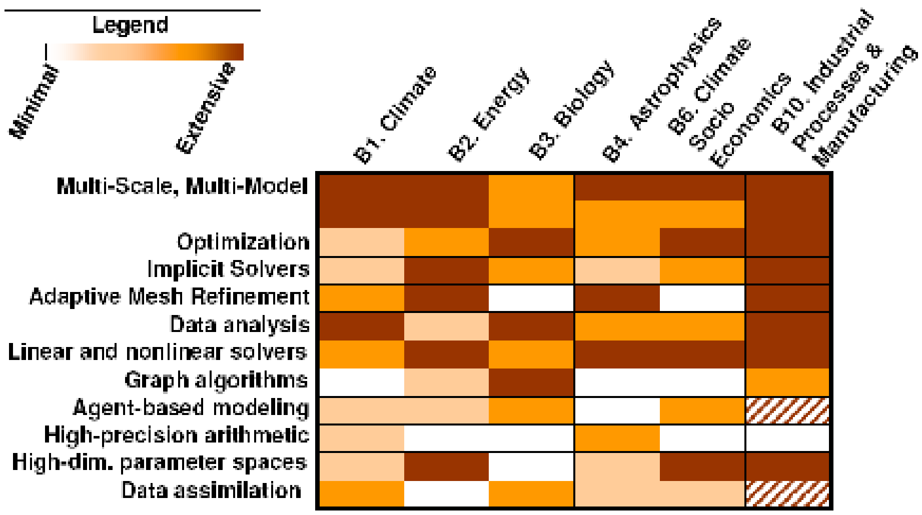

Figure 2: Application analysis from [3]

These figures are included for information and as a possible model for how we may collate application and algorithm inter-dependency information within this roadmap. We will take similar approach with the roadmap and use the information gathered to date to create a similar mapping. The applications that have been included in workshops to date include:

- Gas Turbine CFD Applications
- Cosmological Hydro Simulations for Galaxy Formation
- Multiscale Mathematical Models of the Heart
- Spin dynamics for cancer diagnostics
- Computational finance
- MD Simulation of Complex Biomolecular Systems: Computational Challenge
- ICOM (Imperial College Ocean model): 3D adaptive unstructured mesh ocean modeling
- Industrial CFD for design (virtual engine)
- Biophysics: membranes with lateral phase separation
- Materials science: species diffusion
- EHL – Elasto-Hydrodynamic Lubricant simulation
- Phase-field modelling (PFM) (solidification/crystallisation of molten metals)
- Chemical diffusion through skin (CDS)

Further input is being gained from
- NERC (Climate, Met etc)
- Acoustics & Electromagnetics
- QCD
- Microfluidic flows
- Astrophysics
- Biology & systems biology
- Large scale agent simulations
- Text mining

Algorithmic Challenges

Annex 2 and 3 provide an overview of the current algorithms that are used in the applications considered and the software that is being used at present. A number of current issues have been identified that include:

Load balancing
- meshes
- particle dynamics and computation of interactions

FFTs and Poisson solvers
sparse and dense diagonalisation (eigenvalues)
sparse and dense linear solvers
Use of novel architectures
- FPGAs
- GPUs
- IBM Cell

Domain decomposition
Coupling between different codes
Meshes
- generation of accurate surface mesh
- partitioning

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

Future requirements that have been identified to date include:

- FFTs
  - More scalable alternatives to FFTs and convolution
- ODE
  - explicit algorithms likely to be favoured
  - optimal step-size
- PDE
  - Better preconditioners for hyperbolic and elliptic operators
- Multigrid
  - Algebraic Multigrid (AMG) also as a preconditioner
- Meshing
  - Adaptive meshing
  - partitioning techniques for adaptive and moving meshes
  - good standard for mesh input/output
- Adjoint technologies for data assimilation and sensitivity analysis
- Sparse solvers
  - Direct - new, more efficient methods
  - Iterative
    - Better parallel preconditioners
    - Block Krylov methods
- R-Matrix technologies
  - splitting into inner-outer regions matching at interface
- Arnoldi propagator toolbox
- Partitioning and domain decomposition
  - better partitioning algorithms
  - bandwidth reduction
- Diagonalisation
  - sparse
    - block Krylov methods
    - better Davidson-like algorithms
  - dense
    - nonlinear problems
    - more scalable algorithms
- Fast Methods for dense matrices
  - H-Matrices (Herarchical Matrices)
  - FMM (Fast Multipole Methods)

BLAS - efficient parallel BLAS (PBLAS)

Optimization

- 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
- Good bounding strategies
- Warm-starting
- Semi-definite optimization (state of the art is small, systems are inevitably dense)
- Better techniques than BFGS for molecular geometry optimization

ALE - Arbitrary Lagrange Eulerian

Particle dynamics

- Better solution for mesh mismatchin for long-range interactions

In report [3] a number of key algorithmic issues were also identified. These included research into a new generation of algorithms that

- repeat calculations to avoid storing intermediate values, to minimize movement from DRAM
- will have appropriate mechanisms to allow fault-tolerance and resilience
- take account of the amount of power that will be consumed
- allow mixed precision computation
- include error detection and correction

and of course we need to have algorithms that will adapt to the heterogeneity of the system architecture.

Roadmap Elements

General Considerations

A number of themes have been developed through the consultation. Not surprising these are largely mirrored in other international activities although some are local to the UK. We note particularly that the general themes that have emerged appear to match those in the French activity "Thinking for the Petaflop". It is important that we focus on understanding and considering the UK areas of strength within the context of these themes to make sure that investment and development build on them.

UK Strengths

The 2004 International Review of Mathematics [11] 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. In the workshops other areas were identified where groups in the UK are making international contributions these include multi-scale and multi-level problems, approximation and neural networks.

These are the areas of strength in numerical analysis; we need to consider further afield regarding relevant computer science and application strengths.

Cultural

- Cultural issues around sharing
  Some application domain scientists are used to sharing models and codes, and reusing other people's software. For other domains this approach is almost completely alien with codes being entirely developed within a particular group and little use being made of libraries or other third-party software.

- International boundaries/collaborations
  Many of the application groups have international collaborators or in some cases depend upon software developed in other countries (particularly the US) that may or may not continue to be supported. It is suggested that a map of international developments is created and a repository of information about ongoing activities is developed.

- Development of a Community
  There was a general desire to have activities such as this workshop to develop more of a community across applications and across application/numerical analysis and computer science borders. Bringing together these interdisciplinary groups is very valuable and allows a transfer of knowledge from one field to another. A sequence of events and activities should be developed to assist in the communications across the community.

Applications and Algorithms

- Cross application commonality
  Annexes 4 and 5 bring together the algorithm and software requirements for the applications considered. We try to quantify this later in the roadmap.

- Integration across models
  Many applications involve multiple models at different scales or for different elements of the application.

- Integration of application pipeline (e.g. CAD, simulation, Visualisation)
  In many applications there is a pipeline of activities: first, setting up the model; then the actual calculation; finally visualisation and analysis. A common concern was the lack of integration of the pipeline thus requiring a lot of effort to go, for example from the calculations/simulations to the analysis.

- Error propagation across mathematical and simulation models
  It was recognised that there is a great deal to be understood regarding error propagation through a given model. This is compounded in the integration across models and pipelines.

- Adaptivity
  There is a need to have adaptive algorithms to adapt to problem characteristics and also architectural constraints.

- Scalability
  As noted above scalability is a huge problem for many application areas, and a desire for all application areas. The desire to solve bigger problems faster is one of the main drivers of this community. Most applications do not scale beyond a few hundred processors, and this is widely perceived as inadequate as we move to petaflop-scale machines.

- Partitioning & Load balancing

❖ Scalable i/o
Input and output is important for applications not just in terms of writing out results but also in terms of enabling efficient and effective checkpointing. As applications scale to larger number of processors, this capability will become increasingly important.

❖ Exemplar applications
It is suggested that baseline models for a set of specified applications are developed to enable communication and benchmarking of new algorithms.

Software

❖ Language issues
A variety of languages are used for application development. There is a need to consider how best to support this mixed language environment to allow better code re-use.

❖ Ease of Use
Higher level abstractions should allow application developers an easier development environment. The provision of efficient, portable "plug-and-play" libraries would also simplify the application developers' tasks.

❖ Support for development of software libraries and frameworks
More effective code reuse is essential. This could be achieved by supporting software library development and frameworks for reuse.

❖ Validation of software and models
There were concerns from many application developers that there are not well defined methods and techniques for validating scientific software and the underlying models. In some application areas observational data can play a role in validation, but for many this is not the case.

❖ Software engineering
It is often the case that application teams developing scientific software are not as skilled in software engineering as would be desired. Guidance on best practice for software engineering development would be a step to assist the community.

❖ Lack of standards

❖ Active libraries & code generation
In order to be able to move from one platform to another it would be beneficial to have underlying libraries that "do the right thing" for any given platform. This is becoming increasingly important with the plethora of new architectures that need to be considered.

Sustainability

There is general concern regarding the sustainability of application codes, software libraries and skills (we consider skills in the next section).

❖ There is a need to develop models for sustainable software that might include
  - Long term funding
  - Industrial translation
  - Open community support

- Other

Knowledge Base

- Lack of awareness of existing libraries/packages
  It became clear through the workshop that there is patchy awareness of what is already available. It would helpful to the community to develop mechanisms for collecting information on existing software and tools and disseminating effectively.

- Skills and training
  All presentations at workshops mentioned skills in academic research groups and industry alike. There are simply insufficient students being trained with the required skills, mathematical, software engineering and high-performance computing. Approaches to this include MSC and graduate training, computational science internships and short courses or summer schools.

- Lack of awareness of expertise
  Providing a repository of expertise of numerical analysis and application domains in the UK may assist in developing appropriate teams for activities.

Future Plans

A workshop in London on January 26th and 27th will reflect on the roadmap to date and should lead to a second version. Input is being sought from missing application areas by individual approach and discussion.

A new element in community engagement is the APACE site that is under development. It will provide the capability for anyone in the community to put together models of their application using an ontology of algorithms, to provide information about algorithms and software as well as application insights. If we are successful at gaining community input through this mechanism we will also be able to extract geographical information about expertise as well as knowledge about the measure of requirements for particular algorithm and algorithms areas.

As we gather more measurable information on application and algorithm use and requirements we will develop the roadmap in a matrix form as above but likely with a further axis regarding parallelization/architecture , i.e. availability on gpus, fpgas and the like.

We hope through the final workshop and the other information channels that we will be able to create a roadmap that will prioritize activities along a suitable timescale.

Conclusions

This document captures the input gained to date by the roadmapping activity and provides an initial view of those areas that will form the basis of the roadmap. Further input is required to identify the priority areas and to determine which of those areas it would make most sense for the UK community to focus.

It is clear that a key component to success will be in bringing together the groups who have been engaged in this activity to form a community where a greater communication across disciplines is enabled.

We expect in the next six months that this document will have gone through several generations as the information gained increases through individual discussion and the community website.

References

1. Report to the President: *Computational Science: Ensuring America's Competitiveness*, President's Information Technology Advisory Committee, 2005.
2. A Report of the NSF Blue Ribbon Panel on Simulation-Based Engineering Science, *Revolutionizing Engineering Science through Simulation*, 2006.
3. ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems, www.sdsc.edu/~allans/Exascale_final_report.pdf, September 2008
4. DOE Report on Modeling and Simulation at the Exascale for Energy and the Environment, http://www.sc.doe.gov/ascr/ProgramDocuments/Docs/TownHall.pdf, June 2007
5. US DOE SciDAC activity - http://www.scidac.gov/missionSD2.html
6. 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, Dec 2007
7. Mathematics at the Interface of Computer Science and Industry, the Smith Institute for Industrial Mathematics and System Engineering, 2005.
8. Science and Innovation Framework (2004-2010) http://www.hm-treasury.gov.uk/spending_review/spend_sr04/associated_documents/spending_sr04_science.cfm
9. P. Colella, "Defining Software Requirements for Scientific Computing," DARPA HPCS presentation, 2004.
10. Frontiers of Extreme Computing 2007, Applications and Algorithms Working Group, http://www.zettaflops.org/fec07/presentations/Thursday-1130-Zetta-apps-r7.pdf, October 2007
11. International Review of Mathematics, http://www.cms.ac.uk/irm/irm.pdf, March 2004
12. International Review of Research Using HPC in the UK, http://www.epsrc.ac.uk/CMSWeb/Downloads/Other/HPCInternationalReviewReport.pdf , December 2005

Contacts and further information

Issues and input to this Report

Dr Mark Hylton:	Mark.Hylton@oerc.ox.ac.uk

General input to Activity

Prof. A. E. Trefethen, OeRC, University of Oxford	Anne.Trefethen@oerc.ox.ac.uk
Prof P. V. Coveney, University College London	P.C.Coveney@ucl.ac.uk
Prof N. J. Higham, University of Manchester	Nicholas.J.Higham@manchester.ac.uk
Prof I. S. Duff, STFC, Rutherford-Appleton Laboratory	Iain.Duff@stfc.ac.uk
Project website	www.oerc.ox.ac.uk/research/hpc-na

Annex 1: Methodology for this Roadmap

Many applications share a common numerical algorithmic base. We aim to capture the elements of this common base, identify the status of those elements and in conjunction with the EPSRC Technology and Applications roadmapping activity, 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 roadmapping activity consists of a set of subactivities:

1. Ground work: Evaluation of current situation (general activity)
2. Detailed discussion of numerical aspects, current and future in specific applications (workshop 1)
3. Definition of a road map for future development (workshop 2)
4. Iteration on Road map definitions and requirements (workshop 3)
5. Community Engagement

Each of the following three sections will briefly deal with each of the activities. Finally, the last section will introduce some ideas which could be of use in classifying numerical algorithms and the applications they are used in.

The Access Grid and other Web-based technologies could naturally be used to widen the attendance and interaction to wider audiences. The establishment of a collaborative forum, easily accessible from the Web or other means should also be seen as a primary target of this work.

Ground work

Some initial grounding work will be done to evaluate the current situation.

Application development is an extremely time consuming activity. Many applications have extremely long life spans, certainly much longer than the platforms on which they were initially deployed. That makes the tracking of applications to developing HPC platforms as well as to numerical algorithms a very critical, but an all too often *ad hoc* activity.

This activity aims to elucidate aspects of the current status

- Classes of algorithms used, their importance and centrality in the applications considered
- Current requirements and observed limitations in desired achieving scientific aims. This would, of course, include a study of current numerical and computational performance.
- Analysis of current algorithmic content delivery methodologies, their strength and limitations

Classifying Algorithms and Applications

The algorithmic content of applications should be quantifiable with respect to a number of parameters. This, however, must also consider the context in which algorithms appear. For example, a parallel version of an algorithm may be required in one application, while multiple versions of the equivalent serial algorithm running concurrently may be employed in a different application.

At the same time, the relative importance of the classifying parameters must, of course, reflect the current and future technological trend as well as the suitability of particular numerical solutions for their future deployment.

For example, criteria could include:

1. Range of applicability, in the sense of how widely distributed these algorithms are across the applications considered, and how important they will be for any future developments.
2. Scalability. This should be seen both as a function of performance vs. number of processing units as well as a function of required performance vs. problem size.
3. Parallelism type
   - Pure message passing
   - Pure SMP
   - Hybrid (message passing + SMP)
   - Flexible (any of the above, reconfigurable at will)
4. Memory access properties. In most current architecture, data re-use carries a considerable performance premium given the layered structure of memory access. This situation is unlikely to change in the near future, at least for most architectures being developed.
5. I/O vs. computationally-bound algorithms.
6. Existing equivalent/alternative algorithms and their properties.
7. Deployment issues
8. Complexity of development
9. Testing
10. Ease of use across several applications
11. Language and other computational issues
    - Development language
    - Ease of cross-language usage
    - Delivery: package/library structure

A careful analysis of the outcome of this classification would allow a more focused approach to the roadmap.

Workshop I: Applications

A tightly focused activity is required to analyse individual applications or field requirements in detail: a simple survey would not suffice. For a chosen set of applications, we will bring together scientists and numerical analysts to examine in more details:

- Current numerical and computational performance required
- Forecasted numerical and computational performance required to tackle future problems of interest
- Algorithms needed, if known, and their characteristics
- Numerical capabilities required, otherwise, in order to map these to existing algorithms or help the design of new ones
- Current and required algorithmic deployment vehicles (i.e. packages, libraries, etc)
- Mapping to advanced HPC platforms

Workshop II: An initial Roadmap for Future Algorithmic Development

The previous activities should have provided some insight on the distribution and relative importance of algorithmic content across various applications and, in greater detail, on the specific needs for algorithmic development for specific applications. Together, these two views could help in defining a roadmap or strategy to help better to address the evolving requirements and an initial roadmap will be developed. A second workshop will be held to provide a numerical focus on the initial workshop bringing together technologists and numerical analysts both from the UK and abroad. The aim of the scheme would be to have a defined strategy with respect to:

- Common numerical components
    - identification and prioritisation
    - performance and deployment requirements
- Critical algorithmic development areas
    - Maximum impact
    - Maximum importance across applications
- Knowledge "centres"
    - People and their areas of expertise, algorithmic areas and
    - Application areas sharing common needs
- Mapping algorithms to future HPC platforms
- Vehicles for algorithmic deployment
- Algorithmic repositories and accessing these

The initial roadmap was produced by this Workshop.

Workshop III: Final workshop and roadmap review

A final workshop bringing together the initial set of applications and numerical analysts will focus on the draft roadmap and review from the applications point-of view to ensure that it meets the requirements and provides solutions for areas that have roadblocks at present.

The roadmap will be presented and through discussion, breakout groups etc we will provide a direct input for the final roadmap development.

Community Engagement

Community engagement is a key factor. Throughout the activity, Wikis, Web-based forums, email lists, and the like will be used to maximize the exposure of these activities in the academic as well as industrial/commercial worlds, to capture input from a broader community and expand the range of applicability of the findings.

Annex 2: Information on Existing Software

Packages Used or Mentioned		Used?
ADF	DFT (Density Functional Theory) for molecular electronic structure	Y
ALBERTA	Adaptive Hierarchical Finite Elements Toolbox	Y
Castep	Molecular Electronic structure, plane wave basis set	Y
CENTORI	CFD for Plasma physics	Y
CRYSTAL	Molecular Electronic structure, Gaussian basis set	Y
DESMOND	Molecular Dynamics (MD) - ISV proprietary	
DL-POLY3	Molecular Dynamics (MD)	Y
EPOCH	Particle-In-Cell (PIC) code	
FLASH	Eulerian hydrodynamics for astrophysics (galaxy formation)	Y
FLUENT	CFD – uses FV discretisation	
FLUIDITY	CFD - general purpose multi-phase CFD code (oceanography)	Y
GADGET-2	Particle dynamics model for astrophysics (galaxy formation)	Y
GAMESS-UK	Molecular Electronic structure, plane wave basis set	
Gaussian	Molecular Electronic structure, Gaussian basis set	Y
GKW	Gyro-kinetic code for plasma physics	Y
GROMACS	Molecular Dynamics (MD)	Y
GS2	Gyro-phase fluid model package for plasma physics	Y
HELIUM	Time-Dependent Schroedinger Equation for two-electron systems in laser	Y
HYDRA	CFD used at Rolls-Royce	Y
Kalos	Vlasov code	
MADNES	Oak Ridge project: molecular SE recast as integral equation (under development)	
MCNP	Monte Carlo neutron transport (reactor safety)	
METIS	Graph partitioning (reordering for sparse matrices)	
Molpro	Molecular Code (Gaussian basis)	
NAMD	Molecular Dynamics (MD)	Y
Netgen	Mesh generator for small-ish problems	Y
ORB5	Particle dynamics modelling for plasma physics	Y
Osiris	Particle-In-Cell (PIC) code	
PADRAM	Mesh generator	Y
PARAMESH	Parallel mesh generator	Y
ParMETIS	Parallel version of METIS	
PRMAT	Atomic electronic structure code (finite basis set)	Y
SIESTA	Molecular Electronic structure, finite basis set	

Terreno	Meshing for multi-scale avoiding nested grids	Y
TETRAD	Mesh refinement	Y
VASP	Ab-initio Molecular Dynamics (MD)	Y
Parallel Tools		
Data Synapse	Low latency distributed task submission system	
Symphony	(from Platform Computing) - low latency distributed task submission system	

Libraries and Supporting Packages	
ACML	AMD maths library
ARPACK	Arnoldi eigensolvers for non-symmetric (non-Hermitian) sparse matrices
BLAS	Only BLAS from vendors (Intel MKL, AMD ACML) mentioned
CGNS	Mesh information input/output
CHARM++	Communication/relocation layer for NAMD
ESSL	IBM serial maths library (similar to MKL, ACML)
FFTW	FFT
gViz	For interfaces to visualisation framework
HDF5	Parallel I/O
HSL	Harwell Sparse Libraries (Linear algebra library for sparse matrices)
HYPRE	Multigrid
LAPACK	Linear algebra for dense and band matrices (generally from vendors - see BLAS)
Libsci	CRAY scientific library
MKL	Intel maths library
MUMPS	Direct linear solver for sparse matrices
NAG	NAG numerical libraries
OPLUS	Communication layer for HYDRA
PARPACK	Parallel version of ARPACK
Peigs	bisection and inverse iteration for symmetric (Hermitian) eigenproblems
PESSL	IBM MPI-parallel maths library
PETSc	Iterative linear solvers for sparse matrices
Prometheus	Multigrid
ScaLAPACK	MPI-parallel version of LAPACK
SPARSKIT	Serial (non-parallel) sparse matrix solvers

Annex 3: Current Algorithmic Requirements

Current Algorithmic Requirements - 1

- Parallelism
  - MPI (dominant)
  - Multithreading (incl. OpenMP) - very limited use
  - Hybrid/hierarchical - not used
- Multigrid
  - Algebraic Multigrid (AMG)
  - Classical MultiGrid
- Direct solvers
  - dense matrices
  - sparse matrices
- Iterative solvers (Krylov's subspace)
  - CG
  - BiCGStab
  - GMRES
- Poisson solvers
- diagonalisation
  - dense eigenvalues
    - tridiagonalisation
      - QR algorithm (lack of parallel performance)
      - DC Divide-and-Conquer)
      - MRRR (Multiple Relatively Robust Representations)
      - bisection and inverse iteration
  - sparse eigenvalues
    - Davidson (Jacobi-Davidson)
    - Davidson-Liu
    - Symmetric subspace decomposition
  - SVD - dense and Lanczos (sparse)
- Preconditioners
- FFT
- PDE discretisation
  - FD (Finite Difference)
  - FE (Finite Elements)
  - FV (Finite Volume)
- Spectral methods (rare in all application areas at HPC/NA)
- Meshes
  - structured and unstructured
  - adaptive and adaptive refinement
- Domain decomposition
  - mesh partitioning
  - domain partitioning for particle dynamics

Current Algorithmic Requirements - 2

- ODE (mostly time-marching for PDEs)

explicit Runge Kutta 2nd to 4th order
implicit for stiff cases (unspecified techniques)

Arnoldi propagators for TD-Schroedinger equation

Particle dynamics

explicit short-range interactions
approximation for long range interactions (Ewald sum, FFT, etc)
Verlet algorithm

Adjoint methods

data assimilation
sensitivity analysis

Monte Carlo and quasi-Monte Carlo methods

Stochastic differential equations

Random Number Generators

Currently, from standard numerical libraries (MKL, ACML, NAG)

Optimization

BFGS (Broyden-Fletcher-Goldfarb-Shanno) method (molecular geometry)

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

Annex 4 : Future Requirements

General Future Requirements

Much bigger problems
- high scalability essential
- much better load balancing
- performance overall issue
- much larger data set sizes

Parallelism
- Hybrid parallelism: DMP & SMP
- Hierarchical parallelism to map multi-level approaches
- increase modularity: separation of computation and communication
- Parallel I/O
- Efficient one-sided communication
  - MPI-2 inadequate
  - Global array technolgies
- Libraries abstracting multi-core architectures

Hardware
- better use of multi-core technologies
- GPUs and other novel architectures
- Automatic mapping of algorithmic content to hardware/system
  - software cycle >> hardware cycle
- Vectorisation
  - better use of SSE on Intel etc
  - other forms of vectorisation less useful (on the wane, in general)

Error Analysis
- Analysis of particular algorithms
- Sensitivity/uncertainty analysis for problem
- Error propogation across coupled models
- Considerations of single vs. double vs. higher precisions, especially with GPU implementation

Coupling of different codes
- APIs?

Multi-scale problems

Multi-physics

Better training
- Tackling current dearth of HPC/NA specialists

Physics consideration to drive problem size reductions

Integration with post-processing and visualisation
- standard interfaces for visualisation and analysis software

Legacy provisions

Improved validation and verification

Long term managed support for libraries