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

Workshop 1 Report

November 2008

EPSRC

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

Contents

Preamble

This report provides an overview of the first workshop of the HPC/NA Roadmapping activity. It includes all information pertaining to the workshop together with the initial outcomes and thoughts around the roadmap. The report is organized such that all outputs and likely outcomes are reported in the body of the report together with the plans going forward and the initial roadmap; all supporting material is attached in annexes.

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 26th/27th 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, 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 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 1

Seventeen application areas were discussed by forty three workshop attendees over three days. The agenda for the workshop is provided in annex 1 and the attendee list in annex2. The attendees came from a mix of academic institutions, research laboratories and industry.

The workshop provided short presentations by application experts with break out groups to consider the questions provided on a Proforma (Annex 3). In this report we have attempted to bring together all pertinent suggestions and ideas from the presentations and from the discussion groups. A synopsis of all the presentations has been provided in Annex 7 that identifies the key messages from the presentations.

Common Themes

A number of common themes/issues and desires emerged from the presentations. These can be characterized in the following manner:

1. Cultural

- Cultural issues around sharing

  Some application domain scientists are used to sharing models and codes, of 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.
- Need to consider 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.
- 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.

2. Applications and Algorithms

- There is a lot of cross application commonality

  Annexes 4 and 5 bring together the algorithm and software requirements for the applications considered at the workshop most of which come from multiple application areas (this would be good to quantify).
- 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 (appropriately conservative)

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

Scalability is a huge problem for some 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, 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.

3. 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 it would be desired.
- 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.

4. Sustainability

There was general concern regarding the models for sustainability of application codes and software libraries. This is characterized by the two bullets below but essentially there needs to be a better understanding of the options for sustainability. One of those paths, the translation of research codes to industry codes, needs further support.

- Support for sustainability of software
- Translation to industry code

5. 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.
- Skills and training

All presentations mentioned skills in academic research groups and industry alike. There are simply insufficient students being trained with the required skills, mathematical, software, high-performance computing.

Algorithms Identified and challenges for the future

A full list of future algorithm requirements is given in Annex 6. The general algorithmic areas include:

- Scalable FFT
- Mesh refinement
- Eigenvalue/eigenvector (all or few)
- Optimisation
- Iterative & implicit solvers
- Visualisation
- High-performance one sided communications
- Out of core algorithms to enable larger problems

The major issues for the future were seen to be:

1. Load balancing
    - meshes
    - particle dynamics and computation of interactions
2. FFTs and Poisson solvers
3. Sparse and dense diagonalisation
4. Sparse and dense linear solvers
5. Use of novel architectures (in the immediate future)
    - FPGAs
    - GPUs
    - IBM Cell
    - Clearspeed
6. domain decomposition
7. Coupling between different codes
8. Meshes
    - generation of accurate surface mesh
    - partitioning

At the next workshop we hope to determine where these lie on the roadmap in the context of existing activities and further applications input.

Further Application Areas and Other Requirements

Although we were able to include a broad set of applications in this workshop we have identified some key application areas that have not been captured here. These include:

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

- Data mining and statistics
- Digital signal processing (compression)
- Complex networks
- Medical sciences
- Genome Sequencing

There is also a need to make a collection of information on existing software libraries and activities across the application areas and numerical software.

This workshop did not include any vendors' presentations or inputs. As we move forward it will be important to bring in vendors to ensure that we understand their development directions both in terms of hardware and software environments.

It was suggested that model application codes that can be used as a baseline by algorithm developers would be helpful in providing an effective collaborative framework.

Initial thoughts on Roadmap

General Considerations

It was agreed that there needed to be a bottom up and top down approach to the development of the roadmap. By that we mean that there is a need to identify a Utopia – where we would like to be in twenty years time, perhaps the Grand Challenge of HPC/NA – and through a bottom up approach capture the landscape, develop the community and provide appropriate milestones toward the Utopia.

The Grand Challenge is to provide

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

Specific Roadmap Elements

Here we identify the themes of the roadmap and as we progress we expect to populate the milestones with specific dates and further details.

1. Cultural
    - Identify potential community players
    - Develop models of community sharing
    - Provide community activities, workshops, virtual meeting spaces.

2. Applications and Algorithms
    a. Identify exemplar applications
        i. Develop baseline models for communication and benchmarking
    b. Develop map of algorithms across application domain
        i. Indentify impact of specific algorithm development across discipline groups
    c. Develop map of developments internationally
        i. Collect information about ongoing related activities
        ii. Discuss with international funding agencies plans

3. Software
    a. Abstractions
    b. Guidance on best practice
    c. Develop frameworks for developers

4. Sustainability
    a. Develop models for sustainability
        i. Industrial translation
        ii. Open community support
        iii. other

5. Knowledge Base
    a. Develop mechanisms for collecting information on existing software
    b. Develop mechanism for continuing community input

Next Steps

There are a number of parallel efforts that will be taken forward following up from this workshop.

- We will approach scientists in the application areas identified above to try to capture the requirements from those domains.
- We will collect further input generated by this report and will begin bringing together a knowledge base of existing software libraries and activities.
- We will reach further to international collaborators to get feedback.

The outcomes to date will be considered at a second workshop to be held in Manchester on December 8/9 2008 where the focus will be on the numerical content of the roadmap.

An iteration of the roadmap will be circulated before Christmas and a follow up workshop will be held at UCL on January 26th and 27th 2009. We will also aim to engage other groups and communities in this time frame and specifically the Collaborative Computational Projects (CCPs) who have a steering group meeting in January.

At the end of this initial round of activities we hope to participate in an EPSRC Town Meeting on the findings of this roadmapping activity and that of the architecture group being led by EPSRC.

Conclusions

The first workshop was very successful in bringing together an initial set of application scientists, numerical analysts and computer scientists. It has provided a base set of information on which we can build. It is clear, however, that there is much to be done if we are to succeed in developing a first instantiation of the roadmap in the next four months. We can only do this by engagement of the wider community and we ask anyone reading this report to get engaged and provide input on the material presented here or indeed any element related to this activity.

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.v.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

Acknowledgements

We are grateful for the support provided by EPSRC for the algorithm/application roadmapping activity. Workshop 1 and the follow up analysis have been successful thanks to the work of the team at OeRC and in particular Mark Hylton and Stef Salvini.

Annex 1: Workshop timetable

HPC/NA Workshop 1: Applications: underlying algorithms and challenges for the future
Oxford e-Research Centre

Day 1: Wednesday 5th November
11:00 Welcome
Introductions & aims of workshop by Anne Trefethen, OERC

Morning Sessions chaired by Iain Duff, STFC
11:15-13:15 **Large Scale Simulation**

- Stewart Cant, Cambridge
- Leigh Lapworth, Rolls Royce
- Wayne Arter, UKAE
- Tom Theuns, Durham

Group discussions
13:15-14:00 Lunch

Afternoon Sessions Chaired by Nick Higham, Manchester
14:00-15:30 **Biomedical Simulation**

- Nic Smith, Oxford
- Mark Sansom, Oxford

Group Discussions

15:30-17:30 **Computation for Physical Sciences Session I**

- Ken Taylor, QuB
- Ilya Kuprov, Durham
- Ivan Graham & Robert Scheichl, Bath

Group Discussions

17:30-18:00 Plenary review & discussion of first sessions
18:30 Pre-dinner drinks, St. Hugh's College
19:00 Dinner, Wordsworth Room, St Hugh's College

Day 2: Thursday 6th November
09:00 Welcome to the workshop & summary by Anne Trefethen, OeRC
Morning Sessions Chaired by Iain Duff, STFC
09:30-10:45 **Computational Finance**

- Mike Giles, Oxford

Group Discussions
11:30-13:00 **Computation for Physical Sciences II**

- Chris Goodyer, Leeds
- Björn Stinner, Warwick
- Ian Bush, NAG

Group Discussions
13:00-14:00 Lunch

Afternoon Sessions chaired by Anne Trefethen, OeRC
14:00-16:00 **Oceanography and Climate Modelling**

- Gerard Gorman, Imperial

Group Discussions
16:30-17:30 Review & discussion of day 2
18:30 Pre-dinner drinks, St. Hugh's College
19:00 Dinner, The Boardroom, St Hugh's College

Day 3: Friday 7th November

09:00 Welcome to the workshop & summary by Anne Trefethen, OeRC

Morning Sessions chaired by Nick Higham, Manchester

Developing a high performance computing / numerical analysis roadmap

09:30-11:30 **Computation for Physical Sciences Session III**

- Tony Arber, Warwick (Physics)
- Paul Sherwood (Comp Chem)
- Cliff Addison, Liverpool

Group discussions

11:30-12:30 Review of workshop and outputs

12:30-13:30 Lunch

Discussion – first thoughts for roadmap

Plans for second workshop, Manchester December 2008

15:00 Close

Annex 2: Workshop 1 Attendees

Name	Affiliation
Andrew Sunderland	STFC Daresbury Laboratory
Anne Trefethen	OeRC
Ben Ralston	AWE
Björn Stinner	University of Warwick
Christopher Goodyer	University of Leeds
Cliff Addison	University of Liverpool
Colin M Roach	UKAEA
David Worth	STFC - Rutherford Appleton Laboratory
Dominic Walsh	Schlumberger
Dr Maziar Nekovee	BT Research and UCL
Dr. Leigh Lapworth	Rolls-Royce Plc
Duc Nguyen	Culham Science Centre, UKAEA
Emma Jones	EPSRC
Gareth Shaw	Schlumberger
Garfield Bowen	Schlumberger
Gerard Gorman	Imperial
Graham Riley	University of Manchester
Iain Duff	STFC - Rutherford Appleton Laboratory
Ian Bush	NAG Ltd
Igor Kozin	STFC
Ilya Kuprov	University of Durham
Jamil Appa	BAE Systems
John Gurd	The University of Manchester
Ken Taylor	Queen's University Belfast
Mark Allan	BAE Systems
Mark Hylton	OeRC
Mark Sansom	Oxford
Matthew Piggot	Imperial
Michael J Gillan	UCL
Mihai Duta	OERC
Mike Giles	Mathematical Institute
Nic Smith	Oxford
Paul Sherwood	STFC Daresbury Lab
Peter Jimack	University of Leeds
Nick Higham	University of manchester
Rob Scheichl	Bath
Stef Salvini	OeRC
Stewart Cant	University of Cambridge
Sue Dollar	STFC Rutherford Appleton Laboratory
Tom Theuns	University of Durham
Tony Arber	University of Warwick
Wayne Arter	UKAEA
Zoe Lock	Technology Strategy Board

Annex 3: Proforma for discussion groups.

HPC/NA Workshop 1: Applications: underlying algorithms and challenges for the future 5th-7th November, Oxford e-Research Centre

Session/Application Area:
Breakout Group:
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
Areas of overlap with other applications
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
Knowledge of existing activities in this area

Annex 4: 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 5: 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)
- Optimisation -
  - BFGS (Broyden-Fletcher-Goldfarb-Shanno) method (molecular geometry)

Current Issues

- Load balancing
  - meshes
  - particle dynamics and computation of interactions
- FFTs and Poisson solvers
- sparse and dense diagonalisation
- sparse and dense linear solvers
- Use of novel architectures
  - FPGAs
  - GPUs
  - IBM Cell
  - Clearspeed
- domain decomposition
- Coupling between different codes
- Meshes
  - generation of accurate surface mesh
  - partitioning

Annex 6: 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

Algorithmic Future Requirements

- 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 Lanczos?
    - better Davidson-like algorithms
  - dense
    - Block Divide-and-Conquer methods (?)
    - more scalable algorithms
- Fast Methods for dense matrices
  - H-Matrices (Herarchical Matrices)
  - FMM (Fast Multipole Methods)
- BLAS - efficient parallel BLAS (PBLAS)
- Optimisation
  - Better techniques than BFGS for molecular geometry optimisation
- ALE - Arbitrary Lagrange Eulerian
- Particle dynamics
  - better solution for mesh mismatching for long-range interactions

Annex 7: Synopsis of Presentations

Session 1 – Large Scale simulations -

Title Gas Turbine CFD Applications and Challenges for the Future

Name and Affiliation Leigh Lapworth, Rolls Royce

Subject

ACARE Environmental Goals for 2020:

- Reduce fule consumption by 50%
- Reduce external noise by 50%
- Reduce NOX by 80%

All development in-house but combustion

Current Packages, Libraries, Tools etc

HYDRA

- CFD Solver
- Steady and unsteady flow
- Hybrid unstructured mesh; moving mesh
- Parallel on DMP and SMP systems
- Linearised, unsteady and adjoint CFD capabilities
- Libraries: OPLUS, CGNS, ParMETIS, HDF5
- F77 mostly
- RANS (Reynolds Averaged Navier Stokes); LES (Large Eddy Simulation)

PADRAM

- Multiblock structured and unstructured mesh generator
- All geometry and meshing parametric

Current Algorithmic Requirements

- MultiGrid (HYDRA)
- Preconditioners (HYDRA)
- Partitioning (HYDRA through ParMETIS)

Future Developments, Requirements, Challenges and Issues

- Virtual Engine
  - Multi-physics
  - Different models (RANS for compressor, LES for combustor; RANS for turbine
- Coupling of different codes (Multicode)
- Industrial CFD code development
  - Much increased number of nodes
- Hardware
  - Better use of multicore: through an API?
- Software shelf life >> hardware shelf life
- Recruitment an issue: not enough HPC specialists available

Title Computational Fluid Dynamics

Name and Affiliation Stewart Cant, Cambridge University

Subject

Review of CFD current state and challenges

- Multi-scale: cannot resolve all scales at reasonable cost: turbulence models required
- Major limitation: all practical flows are turbulent
- Complex geometries
  - CAD data format, cleanup and repair issues
  - Surface meshing accurate (by hand); volume meshing hopefully automatic
  - Visualisation and post-processing
- Main techniques
  - DNS: Direct Numerical Simulation of the Navier-Stokes equations
  - Large Eddy Simulation (LES): actively developed – modelling required at sub-grid level
  - Reynolds Averaged Navier Stokes (RANS) – average the governing equations – model all scales – inexpensive hence standard approach
- Numerics
  - Finite Volume – standard in almost all CFD, second order accurate
  - Finite Differences – for high-order accuracy in DNS
  - Finite Elements – rare in FD (common in structural eng.)
  - Spectral Methods – rare, turbulent research
- Solution Algorithms
  - Incompressible flow
    - Poisson equation
    - Conjugate Gradient, MultiGrid
    - Semi-implicit in time
  - Compressible flow
    - Density-based time-marching
    - Explicit integrators (Runge-Kutta) 2^{nd} to 4^{th} order
  - DNS
    - Explicit Runge Kutta 3^{rd} to 4^{th} order
- Performance
  - Critical issue (e.g. LES)
  - MPI/Linux basic technology

Current Packages, Libraries, Tools etc

Current Algorithmic Requirements

Future Developments, Requirements, Challenges and Issues

- Parallelism
  - Distributed memory: minimise global operations
  - Little scope for vectorisation techniques
  - FFT is dead, Poisson solvers are struggling
- Explicit algorithms are favoured
- Exploit synergy between DNS/RANS/LES
- LES requires a major effort to achieve robustness
- Numerics
  - Adaptive meshing
  - Optimal time-stepping
  - Parallel tools for all tasks: CAD → mesh → solution → postprocessing
- Non-standard processors: Cell, GPU, etc

Title Algorithms for Nuclear Fusion Plasma Physics

Name and Affiliation Wayne Arter, EURATOM/UKAEA, Culham

Subject

Tokamak modelling (MAST,JET,ITER)

Current Packages, Libraries, Tools etc

CENTORI

- Fluid model

GS2

- Gyrokinetic phase-fluid model

ORB5

- Gyrokinetic particle (trajectory) model

MCNP

- Monte Carlo neutron transport (for nuclear fusion safety analysis)

Current Algorithmic Requirements

(Gyrokinetic unless stated)

- FFT (exchange between different meshes) (fluid and phase-fluid)
- Domain decomposition (phase-fluid and particle)
- Matrix splitting, solvers and preconditioning (phase-fluid and particle)
- Particle tracking on mesh (particle)
- Particle tracking/ray tracing through geometry (neutrons)

Future Developments, Requirements, Challenges and Issues

- Good preconditioners for hyperbolic and elliptic operators
- Better FFTs and domain decomposition, multiscale generally
- Visualisation

Title Cosmological Hydro Simulations of Galaxy Formation: physics and numerics

Name and Affiliation Tom Theuns, Durham University

Subject

Simulation of the time evolution of galaxies (formation etc)

Current Packages, Libraries, Tools etc

GADGET2

- Particle dynamics model
- Time-steps depending on particle (saving on force evaluation)
- Domain decomposition
- Compute explicitly forces between nearby objects
- FFT techniques for objects between far objects (approx)
    - Load imbalance; different distribution required than mesh for particles (memory layout issues)
- MPI-parallel
- No multithreading
- Extensively ported
- Load imbalance for systems with large dynamic range

FLASH

- Block-structured adaptive mesh
- Eulerian hydrodynamics
- MPI-parallel
- Domain decomposition

FFTW

- In GADGET2

PARAMESH

- In FLASH

HDF5

- Parallel I/O

Current Algorithmic Requirements

Future Developments, Requirements, Challenges and Issues

- Bigger simulations
- Multi-physics
- Greater dynamic range
- More physics
- Training
- Legacy provision
- Improved validation/verification
- Increase modularity (separate comp. and comm..)
- Standard interfaces (“I/O”) for visualisation and analysis software
- Much increased data sets

Session 2 – Biomedical Simulation

Title Developing Multiscale Mathematical Models of the Heart

Name and Affiliation Nic Smith, Oxford University

Subject

Multi-scale, multi-physics modelling of the human heart

- Coupling of
  - Mechanical heart simulation
  - Heart fluid-flow
  - Electro-stimulation
  - Coronary blood flow
- 15 orders of magnitude between molecular → cell → macroscopic levels
- Analysis → medical diagnosis and intervention in the future

Current Packages, Libraries, Tools etc

- Mostly in-house development.
- PETSc
- MUMPS (direct solver)
- A number of other packages/libraries listed in one slide, but not mentioned otherwise (OpenCMISS, SOFA, LIFEV, OPENFEM: Finite Elements; GIMIAS: visualisation; CMGUI: data assimilation?)
- List of numerical techniques given in the same slides but not mentioned elsewhere (FEM, FD, POD, ALE)
- Paralllelism currently limited to 64 processors? (from talk)

Current Algorithmic Requirements

Future Developments, Requirements, Challenges and Issues

- Validation (data and codes)
- Multi-physics
- Multi-scale
- Load balancing
- Parallel I/O
- Visualisation
- Re-engineering codes and models for new architectures
- Adaptive meshing and MultiGrid to cut down the computing requirements

Title MD Simulation of Complex Biomolecular Systems: Computational Challenges

Name and Affiliation Mark Sansom, Oxford University

Subject

MD (Molecular Dynamics) simulation of complex biomolecular systems:

- Cell membrane transport mechanisms (proteins embedded in lipid bilayers of lipids)
- From experiments, only static structure of proteins – simulation essential
- "particle dynamics" type of approach

Current Packages, Libraries, Tools etc

- GROMACS
- NAMD
- DESMOND (industrial/commercial application)

Current Algorithmic Requirements

- Newtonian physics
- Verlet algorithm for Time-Dependent (TD) integration
- Bottleneck: long range interactions (particularly electrostatic)
- Load balancing (need more)
- Approximation by "clustering" portions of molecules (rather than atomic level)
- Emerging architectures must be considered: GPUs, Clearspeed, Anton from DE Share

Future Developments, Requirements, Challenges and Issues

- Multi-scale modelling: quantum mechanics → MD atomistic (hybrid?) → MD
- Very large systems
    - Load balancing
    - Large amount of data and technologies required for storage and access
    - Large-scale visualisation
    - Multi-level integration: how?
    - Hybrid systems (very difficult)

Session 3 – Computation for Physical Science I

Title Solving time-dependent high dimensional PDEs on MPP architectures

Name and Affiliation Ken Taylor, Queen's University of Belfast

---

Subject

Modelling electron dynamics in atoms/molecules exposed to high intensity laser light to complement laboratory studies. Allows investigation on the Atto-second (10^{-18} s) time-scale, fundamental to electronic motion in atomic/molecular systems. The Ti-sapphire laser is the laboratory "WORK-HORSE" for such studies but complementary theoretical work on just the two-electron atom helium demands the full power of HECToR. The time-dependent Schrödinger Equation describing the particular atom/molecule in the intense Ti:sapphire laser pulse is the high-dimensional PDE.

Current Packages, Libraries, Tools etc

- HELIUM
  - Extremely large memory and data transfer requirements
  - Fortran 90
  - MPI-parallel
  - Finite-difference methods

Current Algorithmic Requirements

- High efficiency on MPP systems (minimum communication overheads)
- Arnoldi propagator for accuracy in time-stepping
  - Extremely large problems
  - Very small Krylov subspace required for time-propagation
- Domain decomposition upper triangular part of two-electron radial space
  - Only nearest-neighbour communication

Future Developments, Requirements, Challenges and Issues

- Very important to widen application to other atoms/molecules, by partitioning space into:
  - OUTER region, where HELIUM two-electron finite-difference methods apply
  - INNER region, where a full multi-electron basis set description is possible and already largely coded for traditional collisional work
- Multi-electron systems would lead to yet bigger problem sizes; INNER and OUTER must be Arnoldi propagated simultaneously and careful load-balancing would be needed over these regions.

Title Spin Dynamics: from sparse matrices to cancer diagnostics

Name and Affiliation Ilya Kruprov, Durham University

Subject

Spin-selection in chemical reactions in the presence of magnetic fields.

- Bird navigation: chemical receptors affected by Earth magnetic field

Ilya made a strong case for physics considerations to make large problems tractable

Current Packages, Libraries, Tools etc

- Gaussian
- ADF

Current Algorithmic Requirements

- Parallel diagonalisation of large systems
  - Eigenvalues
  - SVD
  - Dense and Lanczos (Davidson for smallest eigenvalues) methods
- Restriction of basis set cause very large reduction in problem size
  - Physics driven: not through NA
  - Polynomial scaling algorithm

Future Developments, Requirements, Challenges and Issues

- Standard Arnoldi propagator toolbox for TD problems

Title Future algorithm and software development in industrial applications of PDEs

Name and Affiliation Ivan Graham, Bath University

Subject

Overview of the requirements for complex engineering/biological processes modelling using PDEs

- Often huge ill-conditioned systems
- Additional complications
    - Data uncertainty
    - Multiscale
    - multiphysics

Current Packages, Libraries, Tools etc

- ARPACK
- FFTW
- BLAS, LAPACK

Current Algorithmic Requirements

- Path following techniques
- Adaptive FE for complex flows
- Preconditioners for high Reynolds numbers
- Iterative methods for smallest eigenvalues of large sparse systems (inexact solvers)
- Multigrid
- Parallel domain decomposition preconditioners
- Matrix-free inverse power method
- Monte Carlo methods
- Geometric tracking algorithms

Future Developments, Requirements, Challenges and Issues

- Multi-physics
- Multi-scale
- Algebraic MultiGrid (AMG) preconditioners
- Fast methods for dense matrices
    - H-Matrices (Hierarchical matrices)
    - FMM (Fast Multipole Method)
- Iterative methods for dense system (preconditioning)
- Robust computation of (4D) oscillatory integrals

Session 4 – Computational Finance -

Title Computational Finance

Name and Affiliation Mike Giles, Oxford University

Subject

Overview of current HPC use in the financial world

- Very large number of systems used for financial modelling (centres: New York and London)
- Throughput requirements not capacity computing
- "brute force" approach
    - Limited search for efficient algorithms
    - Buy bigger systems, cost no objection
    - Important metrics
        - Quick deployment
        - Easy modification
        - People expensive, hardware cheap
- Multi-task "trivial" parallelism
    - Large number of independent small serial jobs
    - Handled by specific tools (see below)
- Other parallelism
    - MPI: limited use
    - OpenMP: limited use? (to increase with Multicores?)
- People involved
    - Traders – new financial products through scripting languages
    - "Quants" (quantitative analysts) – many PhD in sciences, develop the models and write codes
- New technologies
    - Not of great interest to quants – limited development
    - FPGAs (of minor interest)
    - GPUs
    - IBM Cell

Current Packages, Libraries, Tools etc

- Data Synapse – low latency distributed task submission system
- Symphony (Platform Computing) - low latency distributed task submission system
- Tier 1 banks
    - Software developed in house
- Tier 2 banks
    - Software from ISVs (Algorithmics, SunGard, SciComp)
- Standard RNG from vendors' libraries (Intel, AMD, NAG)

Current Algorithmic Requirements

- Monte Carlo simulation (60%)
- Finited Difference methods (30%) (currently, no FE)
- Semi-analytic methods involving FFTs (10%)
- Stochastic Differential Equations (SDE)

- Answer to AET's question: optimisation not important
    - Only some classical optimisation for calibration purposes

Future Developments, Requirements, Challenges and Issues

- Increasing complexity of models
    - Increasing importance of MC methods
- Many more calculations
    - More "stress" tests required by regulators, calibrations and sensitivity
    - Costs and power consumption starting being an issue
- Multicores
    - Better use of SSE vectorisation
    - OpenMP or multithreading
- GPUs
    - "Easy" for MC methods
    - Much more difficult for FD

Session 5 – Computation for Physical Science II -

Title Parallel Implementation and Application of Adaptive and Multilevel Numerical Algorithms

Name and Affiliation Chris Goodyer, Leeds University

Subject

An overview of HPC NA activities at Leeds concentrating on three applications

- EHL – Elastohydrodynamic lubricant simulation
  - MPI-parallel
- Phase-field modelling (PFM) (solidification/crystallisation of molten metals)
  - Parallel version is in development
- Chemical diffusion through skin (CDS)
  - MPI-parallel

Also, research into

- Multi-level techniques
- Mesh adaptation
- Adjoint methods

Current Packages, Libraries, Tools etc

- PARAMESH - Parallel mesh generator (PFM)
- Netgen (CDS) - Mesh generator not suitable for large meshes
- TETRAD (CDS) – mesh refinement
- SPARSKIT (CDS) – sparse solvers for serial code
- PETSc
- Metis (sparse matrix reordering)
- gViz for interface to visualisation framework

Current Algorithmic Requirements

- EHL
  - Regular Grid FD
  - High order Discontinuous Galerkin Fes
  - Multigrid – MLAT, FAS
  - Geometrical decomposition
  - MPI-parallel
- PFM
  - FD
  - Continuous FE Solvers
  - Multigrid – MLAT, FAS
  - Implicit, stiff ODE solvers (time)
- CDS
  - FE solver
  - Mesh: periodic, anisotropic, 3-d unstructured tetrahedral

- Global error estimation through adjoint

Future Developments, Requirements, Challenges and Issues

- Better parallel preconditioners (CDS)
- Bandwidth reduction, particularly for periodic domains (CDS)
- Combining DMP and SMP parallelism (multicores)
- Hierarchy of parallelism
- Partitioning and load balancing
    - General not just geometric
- Multi-level algorithms to map onto hierarchical hardware
- Automatic mapping of software to hardware
- Plug-and-parallelise libraries
- Inter-application communication (APIs?) for multiscale multiphysics problems
- Long term managed support for libraries

Title PDEs on Moving Surfaces - Computational Issues

Name and Affiliation Björn Stinner, Warwick University

Subject

Free boundary problems

- CFD: surface active agents in two-phase flow
- Biophysics: membranes with lateral phase separation
- Materials science: species diffusion

Coupled surface + bulk problems

Current Packages, Libraries, Tools etc

- ALBERTA – Adaptive Hierarchical FE Toolbox
- PETSc (in ALBERTA)
- BLAS, LAPACK

Current Algorithmic Requirements

- FE on moving polyhedral surfaces
- Level set approach and phase-field method with unfitted bulk FE
- Gauss-Seidel iteration
- Krylov subspace methods (GMRES, CG, BiCGstab) with preconditioning
- Monotone and classical Multigrid

Future Developments, Requirements, Challenges and Issues

- Efficient parallel BLAS (PBLAS)
- Efficient direct solvers?
- Parallel iterative solvers
    - In answer to question: block Krylov iterative solvers
- Parallel assembling
- Parallel MultiGrid with information about mesh

Title Algorithms In Use In Material Science Applications

Name and Affiliation Ian Bush, NAG

Subject

Overview of some of the main packages used in materials science computation when periodic boundary conditions apply:

- DL-POLY3 – classical MD
- CRYSTAL – Electronic structure
- Castep – Electronic structure

Current Packages, Libraries, Tools etc

- DL-POLY3 – classical MD
  - General parallelisation
  - No external libraries
  - Direct evaluation of short-range interactions, FFT approximation to long range interactions
- CRYSTAL – Electronic structure
  - Gaussian basis set
  - ScaLAPACK
  - Fortran 90
- Castep – Electronic structure
  - Plane wave expansion
  - Potential hierarchical parallelism

Current Algorithmic Requirements

- Domain decomposition (DL-POLY 3)
- General parallelisation
- Dense and sparse (Davidson method) matrix diagonalisation
- BLAS, LAPACK (Castep)
- FFTs (FFTW + vendors' libraries)

Future Developments, Requirements, Challenges and Issues

- Better alternatives or development of FFTs
- Better, more scalable, optimisation methods than BFGS for optimising the structure of materials
- Hierarchical parallelism (all three codes could benefit)
- More scalable methods for the Hermitian matrix diagonalization problem

Session 6 – Oceanography and Climate Modelling -

Title ICOM (Imperial College Ocean model): 3D adaptive unstructured mesh ocean modelling

Name and Affiliation Gerard Gorman, Imperial College

Subject

3D modelling of the ocean circulation, waves etc a the global and local scales

- Multiscale problem
- Need to represent highly anisotropic and complex domains

It aims to develop an open source framework for multiscale ocean modelling

Current Packages, Libraries, Tools etc

- FLUIDITY
  - Open source (LGPL) FE solver for CFD
  - Robust parallel implementation
- HYPRE – multigrid
- PROMETHEUS – multigrid
- PETSc
- Terreno – meshing package for multiscale gridding (avoids grid nesting)

Current Algorithmic Requirements

- FE discretisation
- Adaptive unstructured anisotropic mesh refinement and movement → dynamic load balancing
- Adjoint model for data assimilation and sensitivity studies
- Theta time-stepping
- Linearization by Picard iteration
- Iterative solvers (CG, GMRES) with standard preconditioners
- MultiGrid

Future Developments, Requirements, Challenges and Issues

- Working with adaptive fully unstructured meshes in the vertical
- New numerical techniques:
  - Adjoint
  - Mesh movement
  - Multi-physics
  - Fast solvers
  - AMG (Algebraic MultiGrid)
- Validation and comparison with data and other models (e.g. DYNAMO, MITgcm, etc)
- Adjoint data assimilation and sensitivity analysis
- CGNS for mesh input/output (though better standards would be useful)

Session 7 – Computation for Physical Science III

Title Computational Plasma Physics

Name and Affiliation Tony Arber, Warwick University

Subject

Overview of codes/models required for Tokamak (ITER) and fast igniter fusion modelling (HiPER):

- Gyrokinetic models (GKW) (continuum based kinetic) codes
- Particle-in-Cell codes
- Fluid dynamics codes

Current Packages, Libraries, Tools etc

- GKW – explicit gyrokinetic code
  - FD based
  - Massively parallel (MPP)
- GS2 - implicit gyrokinetic code
- PIC codes (EPOCH, Osiris)
- Vlasov codes (e.g. Kalos)
- No libraries are used

Current Algorithmic Requirements

- Riemann solvers
- FE method
- FD methods
- Lagrangian codes (ALE or remap)
- PIC algorithms
- Domain decomposition

Future Developments, Requirements, Challenges and Issues

- ALE (Arbitrary Lagrangian Eulerian)
- Adaptive mesh refinement
  - Load balancing difficult
- Implicit solvers for stiff parabolic terms
- Robust and scalable matrix inversions for parabolic, linear, implicit schemes
- Modelling QED (Quantum ElectroDynamics) processes
- For PIC: implicit EM field updates
- Improved FFTs on domain decomposed grids
- Replace FFT/spectral methods with FE (full tokamak simulation)
- Stay with Fortran 90
- Mostly known algorithms:
  - Issues are implementation, verification and validation not NA

Title Numerical Algorithms in Computational Chemistry

Name and Affiliation Paul Sherwood, STFC Daresbury

Subject

Overview of current state of computational Quantum Chemistry and bottlenecks

- Classical methods (empirical potentials)
- MD (Molecular Dynamics)
- Ab-initio quantum computations, finite basis set, Density Functional Theory (DFT)

Current Packages, Libraries, Tools etc

- NAMD – MD package relying on CHARM++
- DL_POLY 3 – MD package
- CHARM++ - communication/relocation layer for NAMD
- ScaLAPACK – including some pre-release code for MRRR
- PRMAT – atomic and molecular physics (scattering) code (finite basis)
- CRYSTAL – periodic ab-initio code (Gaussian basis)
- GAMESS-UK – molecular code (Gaussian basis)
- Molpro – molecular code (Gaussian basis)
- Peigs – bisection and inverse iteration
- PLAPACK – QR and MRRR
- Libsci, ACML – CRAY
- ESSL, PESSL – IBM
- SIESTA – molecular code (finite basis)
    - Minimisation rather than diagonalisation
    - Multigrid solver instead of FFT

Current Algorithmic Requirements

- Diagonalisation, partial and complete
    - tridiagonalisation
    - QR algorithm
    - Divide and Conquer
    - Multiple Relatively Robust Representations (MRRR)
    - Bisection and inverse iteration
    - Jacobi
    - Davidson-Liu
    - Symmetric subspace decomposition
- 3D FFT

Future Developments, Requirements, Challenges and Issues

- Block DC methods
- One-sided communication (seen as very important)
    - Global Arrays used in a number of Chemistry codes but issues with portability
    - MPI-2 offers only poor implementations on many platforms
- Poisson equation to replace multi-centre Coulomb integrals (Manby, Bristol)
    - Highly parallel Molpro implementation demonstrated on Clearspeed
- MADNES Project (Oak Ridge) recasts Schroedinger Equation as an integral equation over a grid
    - Multi-resolution analysis
    - Prototype code using wavelet basis
- MD codes require efficient 3D_FFT
- Libraries abstracting multi-core architectures

Title A tale of two applications

Name and Affiliation Cliff Addison, Liverpool University

Subject

Benchmarks for the new AMD Barcelona chip using the Liverpool University new cluster

- xhpl (aka parallel LINPACK benchmark) – dense LU factorisation and solution of equation
- VASP - ab-initio MD (Molecular Dynamics) code

Current Packages, Libraries, Tools etc

- xhpl
  - MPI-parallel
- VASP
  - MPI-parallel
- BLAS

Current Algorithmic Requirements

- FFT
- Diagonalisation

Future Developments, Requirements, Challenges and Issues

- Blocked storage schemes for dense matrices
- Tiling (Jack Dongarra's recent work)
- Recursion (LU, Cholesky, QR)