EXCALIBUR NEPTUNE

CHALLENGES AND REQUIREMENTS FOR MODELLING FUSION PLASMAS AT THE EXASCALE

Joseph Parker, UKAEA

Atomic and Molecular Data Needs for Plasma Applications Workshop

15th April 2021

UK Atomic Energy Authority

ExCALIBUR NEPTUNE
Outline

- ExCALIBUR NEPTUNE
- Motivation
- Physics and HPC challenges
- Project design
- Summary

Introduction

ExCALIBUR NEPTUNE

- NEPTUNE is a new exascale-capable code for high-fidelity tokamak simulations
  - Rare opportunity to design a system from scratch
- Co-design effort:
  - Advanced physics models
  - Exascale hardware
  - Scalable algorithms
  - Uncertainty quantification (UQ) for actionable results
  - Integration with engineering and scientific workflows
- Create a lasting modelling tool
  - Design choices/project structure to enable actionable code, long-lasting project

Motivation

Fusion is in the delivery phase

- New generation of tokamaks, e.g. ITER and STEP
- To simulate experiments and guide reactor design, need high-fidelity, high-performance software
  - Tokamaks need a "digital twin" *in silico*
- Tokamaks are tightly coupled multi-physics systems
  - Core plasma - almost collisionless, fast particles, small fluctuations
  - Edge plasma - collisional, large fluctuations, presence of neutrals and impurities
  - Plasma/wall interactions - heat fluxes, sputtering, reflection and absorption
  - Heating systems - EM waves, fast neutral atoms, absorption in plasma
- Treated separately by separate codes (and communities…)
  - At best, have coupling between codes, but not always consistent
- Codes are quite old ~20-30 years

Motivation

Exascale Landscape

- Exascale machines are becoming available
  - Fugaku (Japan, 0.5 exaflops, CPU only), Summit (US, 0.2 exaflops, GPU accelerated)
- Full tokamak simulations are conceivable (e.g. XGC on Summit)
- Technologies are still emerging - not clear what will prevail
- Need performance portability, ability to perform well across range of HPC systems
- Abstraction layers for parallelisation (e.g. SYCL, Kokkos, RAJA)
  - “Separation of concerns”

Pre-Exascale Challenges

- Current HPC systems are not powerful enough to model full systems of equations
  - Play tricks: we under-resolve, make assumptions (e.g. toroidal symmetry), ignore physics (e.g. impurities, kinetic effects), perform "hero" simulations
- Solvers/algorithms sometimes inaccurate/sub-optimal
- Tendency to focus on model/data agreement, rather than disagreement
  - Should work to de-validate models
  - Requires uncertainty quantification (UQ) infrastructure

These problems are addressed by Exascale systems

... but

- No "standard model" of a tokamak
- Need to learn how to utilise Exascale machines

Example challenge: gyrokinetics with GS2

Solve for the distribution function $g_s = g_s(\vartheta, \sigma, k_x, k_y, E, \mu)$, with self-consistent electromagnetic fields contained in the "gyrokinetic potential" $\chi_s = \chi_s(\vartheta, k_x, k_y)$

Gyrokinetic equation:

coupled to Maxwell's equations, integrals of g_s over E and μ to determine χ_s

- Linear terms contain derivatives in all directions $\Longrightarrow$ halo communications
- Nonlinear term requires Fourier convolution $\Longrightarrow$ memory redistribution in k_x, k_y
- Maxwell's equations require velocity space integrals $\Longrightarrow$ sum all reduce operations

Gyrokinetic-Maxwell system is not trivial to parallelise in *any* direction!

- Real challenge to get this scaling to $\mathcal{O}(10^3)$ cores
- Careful data layout in memory and efficient communications
- Improvements require targeted changes for this specific system
- Optimisation is in tension with need for software to be flexible to model changes

More challenges - a non-exhaustive list!

- Gyrokinetics is just one aspect of the full tokamak system
- On HPC systems, FLOPs are cheap, but communication is expensive
  - Finite difference codes have low operational intensity - not much work per data load/move
  - Models often non-trivial to parallelise efficiency
- Different models
  - (Gyro)kinetic core (5D/6D), fluid edge (2D/3D), hybrid fluid/kinetic models for neutrals
  - Need improved fluid/kinetic models, boundary conditions
- Different grids
  - Structured field-aligned for plasma, unstructured wall-aligned for neutrals
- Anisotropy - disparity in scales parallel/perpendicular to field line
  - Affects gridding, convergence of solvers
- Timestepping
  - Wide range of overlapping scales - what needs to be resolved? Hybrid schemes with corrections?
  - Short timescales in E, B fields $\Longrightarrow$ elliptic solve every timestep (global comms) or sub-lightspeed time steps
- Software engineering
  - Code flexibility, software standards, data management (e.g. XGC >100PB data per day)

NEPTUNE design specification

- Flexible - allow different models, prototyping, simulating sub-systems

  $\Longrightarrow$ modular code design
- Performance portable - quick code, but machine architecture uncertain

  $\Longrightarrow$ Abstraction layer for parallelisation
- Accessible to different classes of user - “black box” users, domain specialists, developers

  $\Longrightarrow$ Domain-specific languages (DSLs), project infrastructure (GUI, docs, wiki, etc)
- Actionable - need error bars on output based on uncertain input parameters

  $\Longrightarrow$ Uncertainty quantification
- Handle complex grids, anisotropy, timescale disparities

  $\Longrightarrow$ Spectral/hp element methods, matrix preconditioning
- Manage contributions from disparate fields of expertise

  $\Longrightarrow$ Separations of concern, development via proxy apps

ExCALIBUR Neptune Community

UKAEA is creating a Neptune community by tendering contracts to area specialists

Plasma theory and code

- Fluid model / tokamak edge code - B. Dudson, P. Hill, D.Dickinson, E. Higgins (University of York)
- Gyrokinetic model for tokamak edge- F. Parra, M. Barnes (University of Oxford)
- Particle in cell methods - B. McMillan (University of Warwick)
- Spectral/hp element methods - D. Moxey (University of Exeter), S. Sherwin, C. Cantwell (Imperial)

Anisotropy

- Matrix preconditioning - S. Thorne, A. Lebedev (Science and Technology Facilities Council)
- Partial differential equations - P. Farrell (University of Oxford)

Uncertainty Quantification and Model Order Reduction

- UQ - P. Coveney, M. Vassaux (UCL), P. Challenor, T. Dodwell, L. Kimpton (University of Exeter)
- MOR - S. Guillas, D. Ming (UCL)

Software engineering

- Domain-Specific Languages (DSLs) - S. Wright (University of York), G. Mudalige (University of Warwick)

UKAEA R. Akers, W. Arter, J. Parker, W. Saunders, E. Threlfall, S. Pamela + others

Design via Proxy Apps

Exascale Project's definition:

"small, simplified codes that allow application developers to share important features of large applications without forcing collaborators to assimilate large and complex code bases"

- Allows separation of concerns
- Components of the code can be studied in isolation
  - Synergies found between different scientific communities
  - ExCALIBUR Benchmarking working group already collecting and studying proxy apps
- Assess performance of algorithms/implementations on different HPC systems
  - Important as we don't know what systems will look like
- Allows us to have a sense of code's performance capability *before* the whole code is written
  - Allows design to be flexible and responsive to problems

https://proxyapps.exascaleproject.org

Summary

- Software infrastructure is crucial to experiments and *in silico* reactor design
- NEPTUNE is an opportunity to design cutting-edge software from scratch
  - designed under uncertainty of physics modelling and Exascale HPC systems
- Gathered a community, pursued separation of concerns, design via proxy app
- Contributions from the Atomic and Molecular Data community will be very welcome!