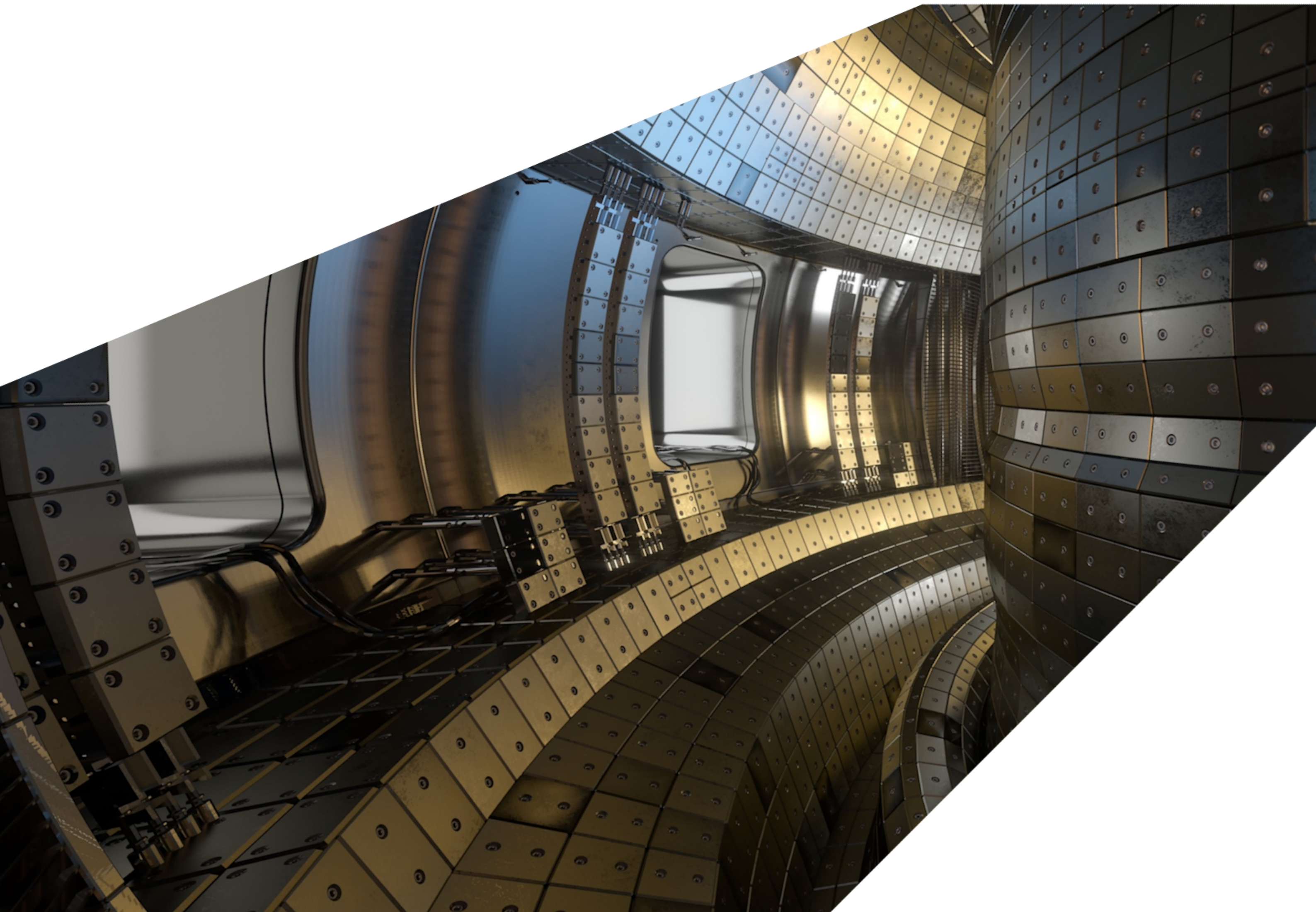
UK Atomic
Energy
Authority

ExCALIBUR

Report of NEPTUNE Workshop 7 October 2021

M1.8.1

Abstract
The report describes work for ExCALIBUR project NEPTUNE at Milestone M1.8.1. It comprises the minutes of the NEPTUNE workshop held on 7 October 2021.



UKAEA REFERENCE AND APPROVAL SHEET

	Client Reference:	
	UKAEA Reference:	CD/EXCALIBUR-FMS/0055
	Issue:	1.00
	Date:	October 21, 2021
Project Name: ExCALIBUR Fusion Modelling System		

	Name and Department	Signature	Date
Prepared By:	Ed Threlfall Wayne Arter BD	N/A N/A	October 21, 2021 October 21, 2021
Reviewed By:	Rob Akers Advanced Computing Dept. Manager		October 21, 2021

1 Introduction

The workshop took place at UKAEA Culham on 7 October 2021, 10.00-16.00 BST. A hybrid format was used with approximately equal split between attendance in person and attendance via Zoom. Attendees were NEPTUNE staff and board from UKAEA as well as external grantees. The workshop had the dual aims of reviewing progress to date and also holding discussions intended to help crystallize future plans, ie. the current position is one of deciding which strategy to pursue, and why.

2 Minutes

2.1 Present

- Wayne Arter, UKAEA *i.p.*
- Michael Barnes, Oxford
- Alys Brett, UKAEA
- Chris Cantwell, Imperial College London *i.p.*
- James Cook, UKAEA *i.p.*
- Peter Coveney, University College London
- David Dickinson, University of York
- Patrick Farrell, Oxford
- Tom Goffrey, Warwick
- Serge Guillas, University College London
- Ben McMillan, Warwick
- Fulvio Militello, UKAEA *i.p.*
- Gihan Mudalige, Warwick
- David Moxey, King’s College London *i.p.*
- Sarah Newton, UKAEA *i.p.*
- Martin O’Brien, UKAEA
- John Omotani, UKAEA *i.p.*
- Joseph Parker, UKAEA *i.p.*
- Will Saunders, UKAEA *i.p.*
- Spencer Sherwin, Imperial College London
- Sue Thorne, STFC
- Ed Threlfall, UKAEA *i.p.*
- Nigel Wood, Met Office

i.p. indicates present in person, otherwise via Zoom.

Additional UKAEA staff were present in a listening capacity.

Note: the slides displayed during the workshop will become available (subject to presenter permission) at https://github.com/ExCALIBUR-NEPTUNE/Documents/meetings/workshop_2021_oct_07 and the agenda at https://github.com/ExCALIBUR-NEPTUNE/Documents/tex/t18. Access to the repository is by recognized name / email only - to arrange access contact neptune@ukaea.uk.

2.2 Morning review session

2.2.1 Wayne Arter - Introduction

First was a review of strategies for software development / engineering, beginning with a 'lite' version of the ECSS-E-40 standard for ESA space applications (ECSS is European Cooperation for Space Standardization) - this derives a *Technical Specification* (TS) to represent what the 'customer' wants. For reference, the basic TS for NEPTUNE is a suite describing the tokamak edge in a comprehensive range of detail levels using a large range of computing platforms (from laptop to HPC), straightforwardly mutable to include new physics, and aiming to provide uncertainty quantification in all cases.

One issue with the customer-driven approaches in the case of NEPTUNE is the disconnect between the providers of the funding - Met Office / BEIS - and the intended users of the end product - UKAEA and other physicists and engineers; it is quite clear that the end users are not the ones providing the funding, so one must be careful in identifying the 'customer'.

Semantic Software Design by Hewitt (2019) emphasizes justification of why the project outputs are good for all customers. Also has useful lists 'how are we going to do this'. Many development ideas in this book are to be applied to NEPTUNE (note there are a number of relevant TLAs that are not summarized in these minutes).

One complication faced by NEPTUNE is the range of users, with level-of-detail needs ranging from laptop simulations up to the Exascale; of the latter, the larger machines are something of a moving target so ensuring up-to-dateness is a challenge.

The project is currently at c.20% spend and $> 40\%$ duration. A slide showed the outputs to date (reports: 40 UKAEA, 25 grantee; proxyapps: 1 UKAEA, 3 grantee); WA stated from his private sector experience that the cost of a report was c.20k - on this basis NEPTUNE is good value for money. He remarked also that there is expected to be significant UKAEA development work (now transitioning from primarily external development to a hybrid of external / internal).

Question from Fulvio Militello re. acronym on WA slide - 'RB' is 'Requirements Baseline'. The impression given was that FM liked the Hewitt-type approach. FM also wanted to see a GANTT chart showing when (presumably code) outputs would appear and at what time user feedback would be sought (note Rob Akers gave assurance via the chat that there exists a GANTT chart with all the deliverables and decision points). WA reminded FM that development at this stage was via proxyapps (Ben Dudson's original strategy) and stated that the proxyapps were intended to increase in complexity starting from a simple baseline - the early proxyapps more appropriate to the DJF stage of Hewitt.

FM expressed a desire to see the overall project road map - WA referred him to the Science Plan (and stated that he had no GANTT charts to hand at the time).

2.2.2 David Dickinson T/NA083/20 Fluid Referent Models

DD gave an overview of his activities.

It was noted that infrastructure had been created eg. Slack channel, github, ReadTheDocs (the latter currently sparse).

2.1 Elliptic solver - vorticity equation. SD1D, Hermes-3 1-D fluids.

The York team has seen a significant restructure with the departure of Ben Dudson and Peter Hill from the project. Some tasks have been shifted in delivery date. York have interacted with PEANO performance analysis workshops, VECMA, and exploring the use of ReFrame (Python regression testing framework). BOUT++ preconditioning - Ben Dudson is still expected to deliver a report in collaboration with STFC.

Task 3 - kinetics. Work proceeding via Python implementations eg. 1-D 2-species Vlasov-Ampère; example seemed to show simple uniform grid outperforming Hermite quadrature approach - higher order not better in this case (Runge phenomenon?). Using Jupyter notebooks.

Task 4 - multi-fluids in 1-D, to be extended to 3-D. Brief mention of design patterns to handle arbitrary number of species and interactions. Showed Hermes-3 design pattern - state object passed down a chain of interaction objects. Issues of inheritance vs. state-command.

All of the above work is still using finite-difference. The intention appears to be that conversion to spectral-hp is the next step.

WA agreed that there are fundamental software engineering issues attached to the multi-species multi-interactions code.

2.2.3 Gihan Mudalige T/NA086/20 Investigate DSLs and code generators

Gihan standing in for Steven Wright of York. He gave a survey of the grant reports eg. systems and their components; software approaches (eg. SYCL a newcomer, also showed instructive synthesis vs. analysis diagram which ranges from general-purpose programming eg. C++ up to domain-specific scientific codes; DSLs are somewhere between the two). Also data representation and I/O: HDF5 etc; also Kokkos Views and Umpire for memory / resource management.

Evaluations of performance portability - BOUT++, Nektar++, EPOCH too large for this to be practical, so resorted to mini-apps eg. MinFE, Nekbone, minEPOCH. Two benchmarks - application efficiency is vs. best yet achieved, architectural efficiency is vs. theoretical max performance. Performance portability results presented, also cited some work done by Bristol (Deakin, McIntosh-Smith). Presented own assessment of SYCL / OP2 / MGCFD using colouring scheme.

GM opinion is that modern C++ best opens up avenues to performance portability. Also note must avoid vendor lock-in risk. Remark that open standards are less agile where the latest hardware is concerned (though note that Intel support for SYCL may help here).

2.2.4 David Moxey T/NA078/20 Progress examining performance of Nektar++ for fusion applications

D2.1 Nektar-Diffusion anisotropic diffusion proxyapp delivered - with github actions and Docker images. Task 1 (meshing) admitted to be somewhat delayed.

DM showed Nektar-Diffusion examples. Initial attempt with step function source poor due to Runge phenomenon (cannot even represent this initial data acceptably well), so substituted approx. step function based on tanh for much better result. Also saw advantage of field-aligned grid.

Showed data layout used for AVX2 kernels: basis function duplicated over multiple AVX2 lanes for vector mult (has C++ wrapper to avoid cumbersome intrinsics syntax, also templated on element shape and polynomial order). Mentioned matrix-free methods. Showed roofline analysis with excellent performance results for straight-sided elements; curved elements significantly slower due to lack of affine property. Advantage from vectorization larger at higher polynomial order.

They plan to do further analysis of performance of the numerical performance of Nektar-Diffusion. Further reporting on Task 1 also imminent.

Questions from Stan Pamela about the code eg. whether using weak form (yes); degree of continuity (C^0 - not using discontinuous Galerkin here).

2.2.5 Peter Coveney T/NA080/20 Methods and toolkits for UQ of high-dimensional parametric and coupled-model applications on HPC

PC opened with general slide on the scientific method and need for UQ. Component-based VECMA toolkit approach. Pilot Job manages HPC workflows by submitting ensemble as 'Trojan Horse'. MUSCLE3 code coupling framework (has been used for eg. transport, equilibrium, and turbulence). US using Radical Cybertools on Summit (not user-friendly). Seemed to be a few problems with QCG-PJ job manager in terms of load-balancing and also a bit of overhead. Note there are quite a few papers on VECMA presenting the new features. Rob noted that STEP project using VECMA to good effect. One anticipated feature is ensemble Kalman filters.

SEAVEA project commencing - fusion is primary use case; showed fast-track (eg. MoGP) and longer-term deep-track components. Noted ARCHER2 machine not yet working to full design scale.

Question from Fulvio about how to handle epistemic uncertainty. PC cited CovidSim where 940 input parameters were whittled down according to significance, with successful outcome; found projected death toll had a heavy-tailed distribution making the mean hard to estimate.

2.2.6 Serge Guillas T/NA081/20 Suitability and potential of reduced order modelling (ROM) to fusion models

SG explained his and Deyu Ming's work using non-intrusive Gaussian process ROM. Obvious 'ghost' defect in surrogate model for anisotropic diffusion but basically within the predicted deviation. Mention made of active learning (AL) approaches and mutual information for computer experiments (MICE) approach with aim to target regions of large uncertainty eg. near turbulent transition. AL helps with loss of space-filling property when a coupled model is used; new sample positions are chosen in a smart way.

Forthcoming AQUIFER project will expand on physics-informed GP (eg. boundary conditions and constraints eg. parameters ≥ 0); also particle filters, and reduced output dimension for models with sharp transitions. Deep GPs can accommodate the latter. Planning to study coupled model of anisotropic heat transport plus isotropic conductor to model 'in-plasma' / 'in-wall' heat transport (note UKAEA's Will Saunders is working on this).

2.2.7 Ben McMillan T/NA079/20 Optimal use of particles

BM summarized his work, much of which had involved providing a version of minEPOCH modified so it was easier to add and test new algorithms. He proceeded to discuss M.3 - implicit particles method. The code solves the Maxwell-Lorentz equations using an implicit method and a second-order Crank-Nicolson stepper to overcome timestep limitations. Jacobian-free Newton-Krylov solver stores only fields in Krylov subspace as particle equations decouple. Iterative method due to large number of DOFs (c.10^9). Error behaviour demonstrated to beat explicit solver. Future preconditioning work continuing even though grant has finished. Also sub-stepping with gyro-orbits resolved for some particles, c.f. Chen et al 2011.

GK-PIC M4/6 incorporated drift kinetics as in ORB5. Control variates reduce noise ($f = f_0 + g$ with f_0 coming from fluid equations).

Future work needed: PIC in unstructured mesh / curved boundary; seed particles tried in Nektar-Driftwave - no conclusions for curved elements or electromagnetic fields from FEM.

Mesh-free methods: PIC/FEM coupling via FCIFEM - MPI issues here. Code is Fortran and requires total rewrite to use SYCL / Kokkos. Plans for EPOCH include submitting independent bid for exascale applications.

WA made it clear that former NEPTUNE grantees are welcome to attend ongoing NEPTUNE project meetings provided that they are prepared to share their NEPTUNE -relevant work.

2.2.8 Michael Barnes, Oxford T/NA085/20 Development of gyro-averaged model

Note UKAEA staff Sarah Newton, John Omotani, and Joseph Parker have all contributed to this work.

Very hard problem, therefore many simplifications involved eg. parallel dynamics only, electrostatic. Starting point is 'standard' drift kinetics. Method of lines and strong stability-preserving

Runge-Kutta. Spectral element code (in Julia and on github) with a higher-order finite-difference check. Wall boundary conditions simplified - ions re-enter as neutrals. Moment-kinetic approach in co-moving frame, no density flow or pressure.

Code outputs compared to analytic solutions; phase-space plots of ion and neutral distribution functions shown. Phenomenology appears qualitatively correct ie. particles ejected from boundary ionize and get accelerated back toward wall. Now extending work to helical, also testing and parallelizing.

WA thanked the grantee, noting very good progress on a challenging problem.

2.2.9 Sue Thorne, STFC T/NA084/20 Preconditioning for NEPTUNE

ST summarized work done Jan-July 2021, starting with an introduction to preconditioning - goal is to transform to a matrix with clustered eigenvalues or with small condition number. Note that a non-symmetric matrix makes iterations more expensive, also convergence may be non-monotonic. Two basic approaches: general preconditioner, or block preconditioner based on physics.

BOUT++ diffusion problem used PetSc solver; explicit matrices so small problems up to 'order' c.8000. Big improvement shown here when preconditioner applied to GMRES (Patrick Farrell asked whether GMRES was restarted - answer was 'no'). Nektar++ ADR solver matrix problem not as successful - small reduction in number of required iterations (however, preconditioned solution was more accurate so should permit a larger time step); todo here is to incorporate MCMCMI. Blocked preconditioners in BOUT++ SD1D, which already has a good preconditioner (audience question from Julia Inca Chiroque revealed that MCMCMI was being used in this case): small improvements shown; not solving to large accuracy as only 10 iterations permitted, but their work again allowed a larger time step. Case D4 with neutrals (with which the BOUT++ developers are themselves not happy) showed good improvement c.35% (further results documented in their reports).

Proposed future roadmap included exploration of state-of-the-art time-advancing techniques, including anisotropy, and coupling continuum and particle representations. Note Rob Akers wanted to talk to ST re cross-cutting exascale computing project.

2.2.10 Patrick Farrell, Oxford: anisotropic diffusion

PF reviewed briefly his work - theory for spectral convergence does not apply in cases of strong anisotropy, hence application of a trick from nonlinear elasticity introducing new variable into problem. Test cases from Deluzet-Narski (asymptotic-preserving). Results showed good performance unless X-point present - difficulty with the latter seemed to be non-smooth solution, known to be bad for FEM.

2.3 UKAEA in-house work

These were all short five-minute presentations.

James Cook showed his work on particles (greatly expanded upon in the afternoon session).

Joseph Parker showed the outputs from the VECMA hackathons eg. applying Gaussian process and machine learning surrogates to BOUT++.

Will Saunders gave an overview of his work with the Soldrake 1-D scrape-off layer model and also work designing a Julia particles framework.

Ed Threlfall gave an overview of simulations of two-dimensional vertical natural convection using Nektar++.

2.4 Afternoon discussion sessions

2.4.1 Software engineering

WA structured this session around some notes by Ben Dudson (who was not present at the workshop); in addition, he revisited four sources of knowledge re s/w engineering: the ECSS-E-40 'lite' standard (successfully used in by WA the past, albeit for projects smaller than NEPTUNE), the text by Hewitt, the approach of Spencer Smith / Dubey et al, and the approach detailed in the textbook *Physically-Based Rendering:From Theory to Implementation* by Pharr, Jakob, and Humphreys (one criticism of the latter is that it is too large, at 1235 pp., for easy assimilation). WA proposed we rule out the following: full-agile development with rigid release dates (releases therefore have much potential for errors / bugs and users are therefore thrust into the role of beta-testers); the customer-supplier paradigm (argued not to be fully appropriate), and the document-driven approach espoused by, among others, Spencer Smith and Dubey et al on grounds that the 'faking' documents *post-facto* seems dubious, and also that Hewitt is overly commercially-oriented. Hewitt is 'defensive' and includes a lot of justification in order to fend off eg. in-house competition.

Of Ben's material: a set of repositories now exists (as recommended) and there is little argument against proposals for testing and documentation. WA noted that the Slack channel is somewhat underused. Re training, David Moxey said Nektar++ training would become available soon (limited by availability of postdoc); also VECMA workshops had occurred.

The project is using github (*excalibur-neptune* repo) and gitlab (the latter UKAEA-internal). WA has some reservations here regarding learning curve and training needs (particularly use of git via command line). Ben defined a standard workflow (which helps): developers contribute by adding their own personal branch which is later merged. Ben's inclination is toward early sharing of code, rather than exhaustive checking by individual devs. Re licensing, MIT is most flexible but most proxyapps use BSD3.

Coupling proxyapps - current situation seems to be that this requires significant development or even rewrite. Associated issues are integration testing, documentation of interfaces, consistent style, clear definition of project aims (cf. Hewitt).

Work has at least started on the following: lists of acronyms and mathematical symbols, choice of physical units (presumed SI for inputs and scaled units internally), index of terms (eg. 'code' - confusion in EuroFusion Road Map where this term applies to software and also buildings regulation), code modularization (at least three layers anticipated), and DSLs. All devs are expected to participate in reviewing and testing inc. unit tests, performance tests inc. periodic regression tests using possibly Amazon AWS / GPU servers. Automation of any of this testing is encouraged, as is the recording of timing information from the outset.

Code style - in the past, this has been one of the most debated areas. Example is `CamelCase` vs `pot_hole` (WA favours latter) name convention; clang_tidy can automate some of this so only an issue for how the code is stored.

Documentation should be dynamic, under version control, and tightly-coupled to the source code ie. embedded or at least in the same directory. Doxygen said to produce nice, browser-friendly output. LatexToHtml also explored - functional if not the smartest in appearance; note help available from the team responsible for the website `https://excalibur.ac.uk`. Markdown recommended for readmes. An additional tool is the pandoc file converter (note that this is also useful for providing a standard for 'non-standard standards' eg. Markdown).

Re DSLs, workshops were held previously (8 Apr, 23 Jul) with the conclusion that there would be a user-interface DSL (Python / Julia favoured over shell script) and a HPC DSL (SYCL or Kokkos / C++ favoured over eg. Cray Fortran). Issue of different levels of user.

Conventions eg. exascale / Exascale (also proxyapp vs. miniapp - solved using Latex newcommand). Tab and line width can be automated. C++20 is likely target language (but issue with some HPC machines not having sufficiently up-to-date compiler).

WA invited discussion on issues of debugging / user scripts, up-to-date version of *CMake*, *Travis*.

Discussion re Doxygen alternatives. PF said Firedrake uses Sphinx which is very good, at least with Python. CC said Nektar++ uses Doxygen and thinks it good with the caveat that it produces large dependency diagrams. JC's view is that Sphinx is newer but nearly identical. DM thinks Sphinx output looks better. WS said Sphinx cannot output xml. WA said having two options here was consistent with the overall NEPTUNE ethos (keep multiple options to avoid possible dead-ends). WS asked what is used by BOUT++ - answer Doxygen exported to Sphinx.

Fulvio Militello queried the rejection of periodic releases - wants release plan in view of forthcoming STEP; also queried conclusion that the customer-supplier approach not appropriate. WA said full agile (inflexible periodic release) only really works if there is a captive user base (whereas NEPTUNE clearly has competition). Customer-supplier model implies that there is a well-established way of achieving target - not the case for NEPTUNE which needs flexibility. FM emphasized that there are some clear non-negotiable requirements which must be specified - WA concurred. FM made the point that the users were UKAEA engineers and physicists and that the code needed to encompass a range of user modes, from full physics to reduced component models; also that the specific questions were well-established (he revealed also a liking for libraries): therefore, he recommended input from Tokamak Science division. WA clarified that there was a requirements baseline. The consensus was that Tokamak Science is part supplier and should be encouraged to give input in terms of physics requirements, but not necessarily code design. JC argued that flexibility in the code could permit straightforward addition of features required in future. FM opined clearly that a bottom-up approach would not lead to success. Changing the subject somewhat, JP came out in favour of at least semi-regular code releases as both good motivation for developers and also from a branding perspective ie. the release allows announcement of any new features or major improvements in the code (ET agreed with this). FM concurred with this also. JC recommended incorporation of tests as part of the release cycle using a benchmark eg. a list of solved problems in edge physics.

JP raised the question of documentation eg. how to write the manual. JC expressed an opinion in favour of the alternative approach of having no manual, just a set of (guaranteed-to-work) examples covering the available modules. Nektar++ does have a fairly comprehensive user manual though its function (in the view of CC) is for reference rather than pedagogical purposes; DM mentioned that tutorials / examples were available in service of the latter (in fact Nektar++ also has a developer manual, though this is currently incomplete and thus rather sparse). Note WA suggests that different levels of user should be exposed to different documentation (cf. Nektar++ user and dev manual) which could perhaps be achieved by clever website design. FM has a positive opinion of the BOUT++ documentation. Further, Nektar++ has notebooks that execute the code in-place (WS mentioned Firedrake tutorials using rst converted into an executable Python script, which improve the user experience by reducing the likelihood of a commit leading to non-working code examples); DM mentioned py.test which runs cells and acts as a regression test. WA said that his students use Jupyter (built on iPython) which can also handle Julia, R, and even C++. DD concurred that this was useful. (Note overlap in the discussion between documentation and regression testing.) Generally important to make sure that documentation and examples actually get implemented, given the wealth of technologies available in this area. One issue, raised by CC, is that NEPTUNE currently uses several different codebases (Nektar++, BOUT++, Firedrake) and thus it is difficult to dictate a unique policy - multiple approaches possible. WA mentioned that the anticipated code structure was a series of layers (ET mentioned the starting position, which was Nektar++-based internals controlled by the DSL of BOUT++). CC noted that the proxyapp used existing libraries and there was relatively little novel code (eg. in the Nektar-Diffusion proxyapp) to which to apply the standards. One significant issue with using up-to-date C++ (eg. C++17) is that the availability of compilers on various HPC platforms cannot be guaranteed. Nektar++ switched to C++ 11 a few years ago (this was in order to remove a Boost dependency, ie. useful features drive updates). Note HPC software stacks are relatively ‘stationary’. One useful C++20 feature is template function speed-ups (WA mentioned also modules; SYCL support also appears relevant). JP suggested containerization / alternate images as a partial solution, though JO noted some HPC requires a machine-specific compiler. DM mentioned that singularity is useful - built for generic architecture. CC suggested that not all users like containers (and noted that Nektar++ has a significant ‘novice’ component to its user base).

2.4.2 Particles

This discussion centered around slides prepared by James Cook showing his work to date and a summary of anticipated issues with NEPTUNE particle simulations. The session provoked lively discussion, which overlapped also with ‘non-particle’ areas eg. fields (which nevertheless need coupling to particles).

John Omotani questioned the need for dynamic mesh adaptivity and noted that in his own work (steady-state turbulence) h-adaptivity was not useful (see Shockwave code). DM raised the question of how to know when a referent model is needed (eg. RANS vs. DNS / LES) - iteration, or trial-and-error? JO said that in his own work, this was known at the start (gyro-radius lengthscale). Perhaps there should be the *option* to dynamically refine. WA mentioned p-refinement error quantification and also local p-refinement. CC mentioned a sensor for local p-refinement - DM noted that this is not perfect; WA asked what was the relevant error indicator - DM answered discontinuity sensor (n.b. not in the sense of discontinuous Galerkin, rather a projection to one order lower). These are local error estimators, not tied to the physics. CC mentioned best to avoid h-adaptivity, but others noted it might be needed in vicinity of X-point.

Methods for suppressing cross-field numerical diffusion were mentioned by Ben McMillan - FCI (flux coordinate-independent) method for higher-order FEM - see his paper *A partially mesh-free scheme for representing anisotropic spatial variables*.

JC proposed orienting basis functions along twisted field lines - twisted grid. BM mentioned that achieving machine-precision quadrature becomes difficult. JC mentioned particles issues in curved mesh - wall intersections - he believes there are software engineering solutions to this problem. Others (WA, JO) seemed less optimistic. WA mentioned also BOUT++ mapped coordinates - can deal with anisotropy problems by elongating elements along field lines (turbulence elongated on this axis).

MPI rank issues - ghost or halo overlap regions with size regulated by $v\Delta t$; mesh adaptivity clearly needs to be coordinated. WA suggested sub-stepping ... big haloes are the worry. JO said electrons easier as they follow the field lines (hence 1-D problem) unlike neutrals. DM mentioned he has worked on a particles code within Nektar++ - tracer particles - and encountered scaling issues (also load balancing if particles cluster); haloes a challenge. WA mentioned probabilistic approach to particle tracking - effectively ray-tracing for neutral trajectories. DM mentioned MPI neighbourhood communicators to minimize global communications, and thinks there is literature on this. JC stated hundreds of particles expected per call, thought most DOFs would be particles; ET cited $(1+p)^3$ DOFs per element per field component, which could well be a significant fraction (possibly even worse for 6-D kinetic implementation?). WA noted few particles / cell was OK for resonant physics - not the case for us, however. CC asked about particles' speed range (he suggested range governed by $\sqrt{\frac{m_e}{m_i}}$, though the range is larger for out-of-equilibrium or runaway electrons). Again, multispecies design issues relevant. WS asked what was typical size of SOL - JO said heat flux width c. few mm, SOL perhaps c.10cm. WA mentioned ITER start up with plasma in contact with wall so SOL width zero - need accurate surface meshing in this case.

The issue of accessing field data came up - projections and non-conforming grids are easy to do in Nektar++ - position in element needed in order to evaluate field (find which element first using eg. octree). Finding which element harder if not planar-facetted - needs Newton iteration as opposed to simple geometrical test. WA mentioned that in neutronics, particle keeps track of what element contains it, but that this would not work well for fast particles. CC recommends substepping. WA quipped that the thing to do was make sure can deal with particles exiting element via the vertex as this is the hardest case. DM suggested backtracking line search back to halo; CC said he thought this must be a solved problem in ray-tracing / computer graphics. DM said field data in halo non-trivial as involves communication - issues updating coupled particles / fields.

2.4.3 Afternoon concluding session

WA asked for any questions concerning the in-house presentations (none) and advertised that our reports are going to be made public (github branch awaiting WA approval).

The main picture re software engineering was one of consensus. There were some important questions to resolve re particles eg. speed limit on implicit PIC; Newton iteration convergence. Ben McMillan said Steven Wright had experience in these (but no present at meeting due to parental leave). SW still involved in NEPTUNE so would be good to arrange a sub-meeting on the particles topic. JC open to input for directing his efforts toward a particle code using FCI / FEM. BM willing to help even though his NEPTUNE funding has expired. WA said another critical thing is whether we can get asymptotic-preserving-type methods working - PF mildly optimistic this will work without need to mesh field-lines, and that he will make available a mini-app written in Firedrake (n.b. this will solve the same problem as Nektar-Diffusion but will incorporate the intermediate-variable trick hence ought to work better).

This was the end of the main meeting, but David Moxey and Chris Cantwell stayed on to discuss Nektar++-related issues with UKAEA staff.

3 Additional inputs from chat transcripts

This is a non-exhaustive summary of the main points made in the chat. These threads had some degree of independence from the main proceedings due to their taking place concurrently with the main presentations.

- Rob Akers / Gihan Mudalige re Pennycook metric

  Question (RA) was whether foreknowledge of UKRI exascale roadmap 2022-2028 would skew use of Pennycook metric for deciding which portability framework to deploy. Answer (GM) was that Pennycook can be used on a downselected set of hardware platforms (as opposed to applying to ‘all hardware’).

- Point about high order methods libraries, Spencer Sherwin

  SS made point that devs of eg. Firedrake need to work closely with PetSc developers (tight integration between solvers and discretization). Patrick Farrell agreed strongly.

- Novel hardware access, Rob Akers

  RA advertised the novel hardware to which we have access via the ExCALIBUR H&ES programme eg. Graphcore, Cerebras (maker of giant chips) - listed on ExCALIBUR website - let him know if access is desired. Also special mention of A100 Nodes / NVLink in Cambridge DiRAC with excellent file access performance.

- Portability framework choice, Rob Akers

  RA made it clear that we have access to 6 FTE of Intel SYCL / oneAPI support via Cambridge Open Exascale Lab and also promise of help from Codeplay. Additional Intel promise of help for deployment on Xe / Ponte Vecchio and Sapphire Rapids. RA has ensured Nektar++ is on Intel priority use case list. Also anticipated alliances with US national labs for access to US exascale machines. Important to know which other libraries etc need to go on priority lists for Intel / AMD / NVidia accelerators. Gihan Mudalige raised question of SYCL adoption - needs discussion with Intel; also Kokkos and OpenMP 5 - Bristol (McIntosh-Smith) able to advise here. Refer to Codeplay for SYCL vs. Kokkos; also Garth Wells promising results - SYCL outperforming native CUDA in FEniCs (n.b. the latter well-written such that porting to SYCL was easy - NEPTUNE take note!).

- Parallellizing sequential design, Rob Akers

  RA asked Serge Guillas whether possible to parallelize sequential design by having each thread do part of parameter space - answer was need to do in a smart way eg. avoid featureless regions of parameter space - but there are some strategies here. Note Peter Challenor (Exeter grantee - not present at meeting) espouses expected-squared leave-one-out (ES-LOO) over MICE.

- Exploiting ExCALIBUR XC coupling project (led by David Emerson), Rob Akers

  RA asked Sue Thorne whether this could be done ... (answer appeared to be affirmative; UKAEA would like to support with c.1 FTE).

Acknowledgement

The support of the UK Meteorological Office and Strategic Priorities Fund is acknowledged.