

ExCALIBUR

D3.1 Software Requirements Specification

TS/1 Response to Tokamak Science Requirements Specification

Abstract

This document provides a response to Tokamak Science's exhaust code specification document. It forms part of the technical specification (TS).

UKAEA REFERENCE AND APPROVAL SHEET

	Client Reference:		
	UKAEA Reference:	CD/EXCALIBUR-FMS/0042/TS/1	
	Issue:	1.00	
	Date:	26 October 2021	
Project Name: ExCALIBUR Fusion Modelling System			
	Name and Department	Signature	Date
Prepared By:	Joseph Parker	N/A	26 October 2021
	Wayne Arter	N/A	26 October 2021
	BD		
Reviewed By:	Rob Akers		26 October 2021
	Advanced Computing Dept. Manager		
Approved By:	Rob Akers		26 October 2021
	Advanced Computing Dept. Manager		

1 Introduction

This document is Project NEPTUNE's response to Tokamak Science's Requirements Specification document. This specification was received by Wayne Arter on 5/11/20 via Rob Akers and its authorship was confirmed to be Fulvio Militello and James Harrison at the NEPTUNE Project Board. It is intended that this document helps to frame expectations of the capabilities of NEPTUNE code, as well as highlighting challenges and areas of potential collaboration.

The requirements are summarized as:

"The new UKAEA Exhaust code needs to be able to capture parallel and perpendicular transport of charged and neutral particles in 3D, full geometry and in a time dependent way. Turbulence should be self-consistently modelled, as well as energy transfer physics between charged particles, neutrals and photons (radiation). While perturbations need to be 3D, a minimal requirement for the code is that it can simulate realistic axisymmetric equilibrium configurations with complex topologies and wall designs. The aim of the code should be to:

1. Efficiently and reliably model exhaust in next generation experiments, like e.g. MAST-U, JT60-SA, and especially ITER (the latter is a stringent requirement for the code).
2. Allow predictive exhaust capability for future reactor relevant machines like STEP or DEMO."

More detailed specifications are given in the following sections either as block quotations or as bold paragraph headers.

1.1 NEPTUNE Science Plan

The Science Plan for NEPTUNE [1] is available online. The stated goal of the project is to:

"develop new algorithms, software and related e-Infrastructure that will result in the efficient use of current Petascale and future Exascale supercomputing hardware in order to

1. draw insights from ITER "Big Data"
2. to guide and optimise the design of the UK demonstration nuclear fusion power plant STEP and related fusion technology

in the approach to the Exascale. The aims of the work are to deliver expertise in, and tools for, "in-silico" reactor interpretation and design."

The Science Plan also describes the software development and theory development that is being and will be undertaken under NEPTUNE.

Software development. The aim of NEPTUNE software development is to provide a flexible framework for implementing different physical models in an Exascale-targeted manner. In particular, the project does *not* envisage a “NEPTUNE system of equations” so much as NEPTUNE providing the ability to solve a class of relevant equations, with models described relatively simply using a Domain-Specific Language (DSL). This flexibility enables the hierarchy of models of varying fidelity and computational costs. It also allows engagement from different classes of user with different levels of physics and software expertise.

Theory development. The above framework approach notwithstanding, NEPTUNE is also supporting theory development in two of its four work packages: FM-WP2 Plasma Multiphysics Model and FM-WP3 Neutral Gas and Impurity Model. These will develop two close coupled models, with FM-WP2 seeking to include kinetic effects in existing and new edge plasma models, and FM-WP3 developing particle-based models for describing the region outside and just inside the plasma (neutral atoms/molecules and partially ionized impurities).

Another work package, FM-WP1 Numerical Representation, is addressing related numerical issues, such as the accurate modelling of highly anisotropic dynamics, the accurate representation of first wall geometry, and the numerical preservation of conservation laws from the underlying models.

As such the NEPTUNE plan is aligned with all the points in the summary requirements quoted above, though there are minor issues regarding the details as discussed in the following sections. It should be noted that the Science Plan outlines a five-year programme that explicitly requires user involvement for fuller development of surrogate models (such as turbulent friction) and to specify the detailed physics of ionisation and excitation reactions. Indeed, development of many of the physics capabilities listed below will benefit greatly from a strong collaboration between Tokamak Science and Advanced Computing.

2 Overall Capabilities

Hierarchical approach with multiple models, going from low fidelity (e.g. laminar fluid and fluid-kinetic runs in 2D) to medium fidelity (e.g. fluid runs with neutrals and turbulence multispecies, but with reduced number of species) and high fidelity (e.g. full kinetic or hybrid kinetic/fluid runs with turbulence and multispecies approach). The plan is for NEPTUNE to provide a framework for solving relevant physics models, and a user-friendly Domain-Specific Language (DSL) for doing this. This allows UKAEA to develop its own physics models in a manner significantly independent of NEPTUNE development and code distribution, see Figure 1. This is the same approach as currently used by UKAEA for BOUT++, where UKAEA has ownership of physics models in the STORM software, while other organizations have ownership of the BOUT++ framework.

Project NEPTUNE intends to provide a suite of examples of physics models as part of the code distribution. As addressed in §3, there is also development of a physics model under NEPTUNE that could be adaptive, displaying a range of fidelities within different regions of the domain during a single simulation.



Figure 1: Development of NEPTUNE software via a sequence of proxyapps.

Numerically efficiency, obtained for example through scalability to large number of cores (given the expected computational resources, the code should be designed to take: ~ 1 week for low fidelity parametric scans; $\lesssim 1$ month for medium-high fidelity runs for advanced design; $\lesssim 3$ months for high fidelity physics studies). Order of magnitude estimates for the cost of code execution (see Annex Section A), indicate that these timings are possible but challenging.

The plan is to target Exascale simulations that can support high-fidelity physics simulations. Project NEPTUNE aims to do this in a performance portable fashion using abstraction layers to separate the physics from the computation details. However, the challenges of Exascale mean that a degree of specialization of the code towards available Exascale machines is expected. At some point this will probably entail trade-offs that are detrimental to performance on smaller machines. Despite this, we still anticipate that the software should be reasonably performant for small-scale runs.

Focus on stability of the code, using (preferably) unconditionally stable numerical schemes and capability to diagnose and re-start failed runs. Ensure a small failed simulation rate for common configurations. The focus will be on accuracy rather than stability. An accurate solution will be a stable one, whereas the converse is not true. The use of bifurcation tracking techniques is expected to help with numerical stability. Moreover, the use of ensemble techniques (as part of uncertainty quantification) should provide information on stability as a function of parameters, as well as making it likely that at least a subset of calculations produces usable results.

In addition, there are some planned technical solutions for diagnosing problems with runs. The simplest functionality is to allow users to change parameters when restarting a job using checkpoint files. The code can also be made to automatically check input files (to catch user mis-

spellings), and to output a file containing the parameters that were actually used in a simulation (in case some were accidentally overwritten). Simulations may also be given a Universally Unique Identifier (UUID) to allow provenance tracking of simulations.

Provide capability to model exhaust physics all the way from sheath limited to strongly detached regimes. This capability follows from (1) the implemented physics models/boundary conditions, and (2) the lengthscales the code can resolve in simulations of feasible run times. Project NEPTUNE anticipates that this will be feasible. See §3 for further details for the physical models.

Modern software design with modular approach (independent and efficient libraries). Project NEPTUNE is certainly adopting a modular approach to software design. This is crucial for enabling many of the desired features, as noted below. The use of reliable third-party libraries is also vital for enabling a small team of developers to leverage the work of others, and to allow code flexibility and ease of prototyping.

Ability to integrate with other codes (e.g. with IMAS). This feature is anticipated, and will be facilitated by writing the code in object-oriented C++.

Integration with IMAS (and other data formats/standards) can be achieved by writing a module to translate between NEPTUNE's internal data structures and IMAS format. NEPTUNE developers are involved in TSVV software which will also integrate with IMAS, so will have experience in this area.

Modern visualization tools Integration with modern visualization tools is not specifically addressed in the science plan. However, this is enabled by using standard data formats such as netCDF/HDF5. Auxiliary tools (similar to BOUT++'s xBOUT library) could also be developed.

The issue of *in situ* visualization will also be important at Exascale, with the need to interrogate large quantities of data without moving it, perhaps during a simulation. Project NEPTUNE does not have specific plans for enabling this. However, the need for this will be widespread, and we expect third-party tools/libraries to become available to support this, particularly in C++.

Accessibility (output easily catalogued and interrogated big data) Given the constraint of producing Exascale volumes of data, it is expected that NEPTUNE will default to using the Met Office practice of only saving the files necessary for repeating a simulation, rather than full outputs. It is intended that key aspects will be captured by surrogates, descriptions of which will be saved and indexed. Users however will be able to choose to output more data and to define custom diagnostics. A modular approach to software design and the use of standard libraries will ease the implementation of these.

In time, there will likely be a move towards creating simulation databases in preference to repeating expensive simulations. Such projects are nascent, but we have collaborators who are working in

this area. Again, the modular approach and use of standard data structures will enable interfacing with such databases as the technology matures.

Version control and user support. Version control is a fundamental part of modern software development, and will certainly be used in NEPTUNE. More generally, the project will use best software practices, including protected branches, peer reviewed pull requests, issue trackers and automated testing. The main code contributors are familiar with such practices; we have heavy involvement from the projects BOUT++ and Nektar++ which have very high quality software practices. Project NEPTUNE also have involvement from UKAEA RSEs.

User support will be provided through issue trackers, a project Slack channel, and where relevant, through training sessions. Project NEPTUNE anticipates supporting several classes of user, from those with limited software knowledge, to users with appreciation of different numerical/algorithm choices, through to numerical/software specialists who might edit the code on a granular level. Support for such a wide range of uses is enabled by a highly modular separation-of-concerns approach.

3 Physics model

3.1 General requirements

Equations need to be applicable to arbitrary aspect ratio devices. The equations employed will not generically make the assumption of small inverse aspect ratio.

The collision operators between plasma components, plasma and impurities and plasma and neutrals have to be sufficiently accurate to properly describe the radiation generated and the energy transfer between species. These operators may be included in the model being derived in FM-WP2, but at this stage simpler model operators are being used.

Photon opacity effects need to be included in the model. The treatment of effects due to radiation are discussed in Annex A of ref [2] (“the equations document”). Initial implementation will account only for usage (1) *the calculation of source/loss terms*, with usage (2) *the production of synthetic spectra*, depending very much on input from experimentalists, as anticipated in the science plan [1, p. 11].

The equations need to be capable of dealing with multiple species in non-trace amounts (at least D, T, He and seeding impurities). The models being derived support distribution functions for multiple species.

The equations should not rely on the Boussinesq approximation. The model derived in FM-WP2 does not use the Boussinesq approximation. Technically, this means NEPTUNE will need to be able to invert three-dimensional elliptic operators with non-constant coefficients. This capability will be available.

The code should evolve both the electron and ion energy (temperature). This is the case with the FM-WP2 model.

It would be acceptable to have only axisymmetric equilibria, potentially corrected through small perturbations. This is answered by the discussion in Section 5.

3.2 High fidelity simulations

“In high fidelity simulations, kinetic/fluid transition for both plasma and neutrals has to be properly captured, potentially using a multi-region approach exploiting different models in different parts of the machine could be considered (for both neutrals and plasma); Boundary conditions need to consider improved sheath physics (collisional &

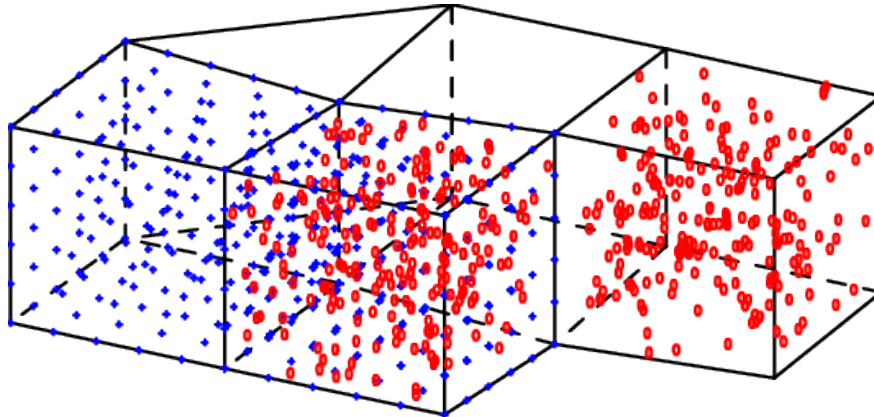


Figure 2: NEPTUNE software will use finite elements and particles to represent different atomic species as needed.

shallow angles); Ideally, the model should be able to treat de-magnetized ions in the divertor region; Electromagnetic effects should be included, but a perturbation approach would be acceptable.”

The Science Plan states that “FM-WP2 and FM-WP3 will concentrate upon development of the two close coupled models of the NEPTUNE programme, specifically FM-WP2 around the inclusion of kinetic effects into existing and new edge plasma models, and FM-WP3 of particle based models for describing the region outside and just inside the plasma (neutral atoms/molecules and partially ionized impurities).” See Figure 2.

In FM-WP2, a novel “moment-based” approach is being developed (for both plasma species and neutrals). This approach is a variant on gyrokinetics (though at present, the derivation exists only for drift kinetics). Instead of the stationary background Maxwellian of the usual gyrokinetics, this approach uses a dynamically-varying background Maxwellian, where the particle density, bulk velocity and temperatures are allowed to vary with time. The result is a hybrid approach, where the software evolves both a fluid system and a modified kinetic system. One of the benefits of this approach is that it gives a natural means to capture the fluid/kinetic transition, by switching between the fluid+kinetic and fluid-only descriptions. Moreover this approach allows the software to detect which of the fluid and kinetic descriptions is valid in any region based on the properties of the modified distribution function, and therefore to automatically evolve the relevant model.

The favourable properties of this approach are valuable, but as with any novel approach there is an inherent risk of the scheme not working. Thus as described in ref [2, § 1.1], in the event that this approach is not feasible, the kinetic calculations will use a Particle-in-Cell (PIC) approach. While full orbit PIC is potentially extremely expensive relative to gyrokinetics, it has been very well-studied both in terms of theory and numerical implementation. Moreover the issue of how to treat the transition from fluid to a particles in a coupled model is well-understood, at least in classical fluid dynamics.

Relevant sheath physics boundary conditions are being derived as part of this work, including shallow angles and collisions. Ions which exit the domain re-enter as neutrals with the Knudsen

cosine distribution.

While the current derivation is electrostatic, there is nothing which precludes the derivation of an electromagnetic model.

3.3 Neutrals and impurities

Different levels of multispecies capabilities should be considered. The framework nature of the code means this will be the case.

In medium to high fidelity models, neutrals and heavy species should be treated fully kinetically (the former have potentially low collisionality and they are not bound by the magnetic field, the latter have a large Larmor radius). Development of a neutral gas and impurity model is ongoing under FM-WP3. Page 7 of the Science Plan states: “Kinetic levels of complexity are . . . necessary . . . for modelling the burning plasma regime, due to the inherent uncertainty in the fluid codes. The plasma in a fusion reactor may well behave significantly differently to plasma in existing devices because it will in general contain two main ionic species (Deuterium and Tritium), neutral fuel particles and ionised Helium ash (or alpha particles), as well as impurity ions originating from the wall.” That is, kinetic models will be derived and implemented, though it is hoped that this will facilitate the development of fluid models.

In high fidelity models, dynamical neutral evolution on the turbulence time scale would be preferable, if possible. High-fidelity simulations will use a kinetic model for neutrals where appropriate.

Ability to model pumps (either directly or via pumping surfaces) should be included. Pellets: the code should enable simulation of, or coupling to models that can model, pellet fuelling. Models for these aspects are not included in the Science Plan. As we will use an unstructured mesh, it will be possible to describe arbitrary pumping surfaces.

The role of sources and sinks is recognised to be crucial in NEPTUNE physics so that modelling of pellets as a source will be possible. The tool CWIPI (Coupling With Interpolation Parallel Interface) may be used for coupling to more complicated pellet models.

4 Physics capabilities

“The code needs to be able to give reliable predictions of:

- Divertor loads:
 - Reliable calculation of upstream particle and heat flux profiles: proper drift physics and upstream turbulence;
 - Divertor turbulence: turbulence spreading in the divertor region, effect of magnetic shear next to the X-point(s) to understand upstream/downstream connection;
 - Multifluid capacity to model radiation and detachment physics;
 - Reliable calculation of the electric field (collisionless physics near the separatrix, proper reflection of neutrals at the target);
 - Able to capture in/out and top/down asymmetries (geometry, drifts, radiation and detachment)
- Wall loads:
 - Should be able to predict filamentary transport and role of hot ion confinement (wall erosion)
 - Should calculate radiation and neutral loads (may require dedicated modules, same for divertor loads);
- Impurity transport, pumping and fueling:
 - Ability to track low (He Be), medium (C, N, Ne) and high Z impurities (W, Ar, Xe) with multiple charge states;
 - Reliable kinetic modelling of neutrals to assess fueling and pumping capacity;
 - Ability to handle non-trace species (D, T, He, seeded species), which requires new closures for the equations (if fluid, beyond Braginskii \implies Zhdanov or better);
 - Accurate model of friction forces (turbulence + neoclassical on open field lines);
 - In high fidelity models, the code should be able to simulate localized gas puffs, via injection of neutral particles.
 - Dust the code should enable coupling to codes to simulate dust generation, transport and ablation, via exchange of fluxes and sources.”

Physics models. Many points in this section relate to the physics model. As we have alluded to above, the actual physics model used is the responsibility of the user; so, for example, it will be necessary for users to derive an accurate model for turbulent friction. Moreover, as NEPTUNE will initially be an edge code, only electromagnetic radiation from the edge plasma will be calculated, and an additional model will be required for radiation for the core plasma. Nonetheless, we plan to facilitate models which have the listed properties. In particular, our meshing approach will allow accurate representation of the walls and divertor. Thus powerful and flexible source/sink models may be used, for both volumes and surfaces.

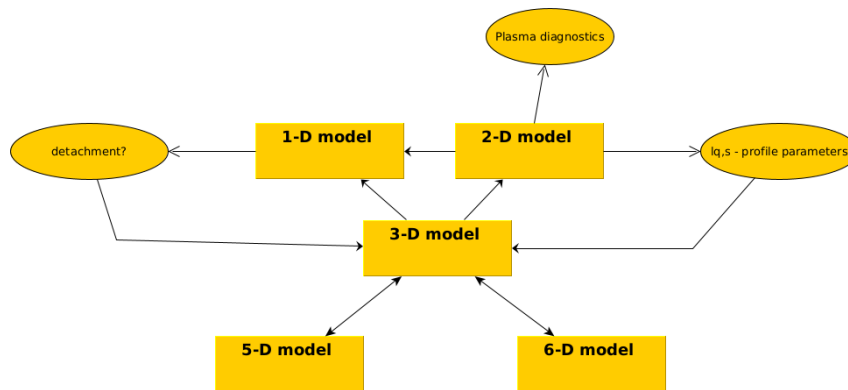


Figure 3: Project software will integrate a range of models of different complexity, here indicated by their dimensionality.

NEPTUNE code will also be capable of representing scales on which filamentary transport has been calculated to occur. The code will only produce fluxes however, and it will be up to the user to calculate wall erosion.

Validation, Verification and Uncertainty Quantification (VVUQ). Naturally, when the solution of a problem is unknown, it is difficult to assess whether the provided answer is reliable as requested in the first line of the current section. Therefore we are integrating NEPTUNE with a suite of tools for VVUQ to enable validation of code outputs against experimental results, and provide a error bars for code outputs due to the uncertainty in the code inputs. As allowed by error estimates, models of different complexity will be used to predict key properties of the SOL, see Figure 3.

5 Geometry

“Ability to handle complex topologies and novel geometries:

- Capacity to treat different topologies with multiple X-points (>2), possibly in a dynamic way, or at least implemented in a way that does not preclude time-varying equilibria being modelled in future;
- Ability to handle singularities in the grid (null points) or to develop accurate scheme(s) that do not have singularities;
- Ability to interface with 3D CAD designs of the machine;
- Capability to handle conformal grids that go all the way to the wall for both neutrals and plasma;
- Capability to treat subdivertor structures (only neutrals).”

NEPTUNE will use a spectral/hp finite element approach. This allows considerable flexibility in meshing, with elemental tetrahedra, hexahedra or prisms being used to describe complex geometries. In addition to refinement of the elemental grid sizes (h -adaptivity), the order of the piecewise polynomial basis functions used on each element may also be changed (p -adaptivity). The production of finite element griddings conforming to CAD descriptions is standard for virtually all meshing packages. Project NEPTUNE will specially aim to use the small number of packages that may produce finite elements with curved edges and faces to exploit fully the higher order basis. This gives a framework which can achieve spectral convergence on complicated, conforming grids, while still being able to handle discontinuities in the solution.

Should it prove necessary to produce a meshing conforming to surfaces of an equilibrium magnetic field, then because the finite element approach does not require a structured grid, it should be able to treat arbitrary geometries and topologies, in particular including multiple X-points. The approach could be extended by further use of adaptive meshing to allow for time-varying equilibria.

A Annex A: Order-of-magnitude estimates for tokamak edge modelling

Suppose plasma number density $n \approx 10^{18} \text{ m}^{-3}$. Order of magnitude dimensions are $L \approx 0.1 \text{ m}$ for SOL thickness, reactor minor and major radii say $a = 3 \text{ m}$ and $R_0 = 10 \text{ m}$, so the volume of SOL $\approx 4\pi^2 a R_0 L \approx 100 \text{ m}^3$. It follows that the total number of electrons $\approx 10^{20}$.

The shortest timescale is inverse eB/m_e , the electron cyclotron frequency, where the ratio of charge to mass for the electron is $|e|/m_e = 1.76 \times 10^{11} \text{ C kg}^{-1}$, and $B \approx 10 \text{ T}$, so $\tau_{Be} \approx 10^{-12} \text{ s}$. Hence the number of particle-steps to evolve 1 s of physical time is $10^{20+12+1}$ (assuming of order 10π timesteps per orbit), which assuming 1000 flop per update, needs a total of 10^{36} flop. Thus to complete a computation in 1 s on Exascale machine, only one 1 particle-step in 10^{18} is allowed. This implies for example only 100 superparticles in the volume can be used supposing each has a weight 10^{18} , which is unlikely to be adequate because electrostatic and other effects will produce a

noise level that swamps any physical effects. However if 3 or 4 months (approx. 10^7 s) are allowed, then a calculation with 10^9 particles may be performed if memory-access bandwidth constraints can be satisfied, when the noise levels may be manageable.

The situation may be shown to be a million times easier for neutrals considered separately, assuming neutral density $n_0 = n$, particle mass m_p and temperature $T_0 = 10$ eV, for a timescale of $\tau_0 = L_{mfp}/v_0$ where the mean free path $L_{mfp} \approx L$ and the neutral speed typically is v_0 . For then $v_0 = \sqrt{|e|T_0/m_p} = \sqrt{|e|/m_e} \cdot \sqrt{m_e/m_p} \sqrt{T_0}$, ie. $v_0 \approx 4 \times 10^5 \times (1/40) \times \sqrt{10} \approx 3 \times 10^4$. Thus $\tau_0 \approx 10^{-6}$ s, ie. a million times longer than the electron cyclotron timescale.

Suppose a fluid model is employed instead, ie. the electron distribution is represented by its first 3 moments. If the electron temperature $T_e \approx 10$ eV, then the thermal speed $V_{Te} \approx 40v_0 \approx 10^6$ m s^{-1} . Supposing the SOL to be sampled at a uniform 1 mm interval, then the number of sample-points $\approx 10^{11}$, and the timestep for an explicit scheme $\approx (10^{-3}/10^6) \approx 10^{-9}$ s, so the number of sample-point updates is 10^{11+9} . Assuming 1000 flop each update, this means one second of physical time needs 10^{23} flop or 10^5 s ≈ 1 d of Exascale machine time, if memory-access bandwidth constraints can be satisfied. If an implicit scheme is used to simulate plasma ions as a fluid on a drift type timescale $\approx L/V_{Ti}$, then possibly the timestep $\tau_i \approx \tau_0 \approx 10^{-6}$ s, ie. a thousand times larger, and calculations lasting only a few minutes might suffice.

Another way of looking at this is to suppose that numerical problem is D -dimensional, 1000 flop are needed for each sample update and N_D samples per spatial dimension and N_D^2 time samples are allowed. Then to update such a model in 1 s requires $N_D^{D+2} \approx 10^{15}$. Thus if $D = 3$, $N_3 \approx 1000$, and $N_5 \approx 100$. It seems that accuracy controlled, unstructured, implicit fluid models should be possible, although for the more complex models 1000 flop per update may be an underestimate by orders of magnitude.

References

- [1] W. Arter, L. Anton, D. Samaddar, and R. Akers. ExCALIBUR Fusion Modelling System Science Plan. Technical Report CD/EXCALIBUR-FMS/0001, UKAEA, 2019. <https://www.metoffice.gov.uk/binaries/content/assets/metofficegovuk/pdf/research/spf/ukaea-excalibur-fms-scienceplan.pdf>.
- [2] W. Arter. Equations for EXCALIBUR/NEPTUNE Proxyapps. Technical Report CD/EXCALIBUR-FMS/0021-1.01-M1.2.1, UKAEA, 2021.