

ExCALIBUR

D3.1 Software Requirements Specification

RB/2 Engineering Requirements Specification

Abstract

This document provides a description of reactor design engineers' requirements for NEPTUNE. It forms part of the requirements baseline (RB).

UKAEA REFERENCE AND APPROVAL SHEET

	Client Reference:		
	UKAEA Reference:	CD/EXCALIBUR-FMS/0054/RB/2	
	Issue:	1.00	
	Date:	November 13, 2021	
Project Name: ExCALIBUR Fusion Modelling System			
	Name and Department	Signature	Date
Prepared By:	Wayne Arter BD	N/A	November 13, 2021
Reviewed By:	Rob Akers Advanced Computing Dept. Manager		November 13, 2021
Approved By:	Rob Akers Advanced Computing Dept. Manager		November 13, 2021

1 Introduction

This specification is based on four main sources, namely

- Presentation by Chris Jones (CJ) at the NEPTUNE internal workshop on 16 December 2019
- Points made in an email by Michael Kovari (MK) dated 19 December 2019
- Interview with Zsolt Vizvary (ZV) on 20 December 2019
- Use by author (WA) of SMARDDA-PFC software in tokamak reactor design 2016-2020.

A significant part of CJ's talk concerned calculations of stress in 'contact' problems and in tokamak-relevant materials, particularly those to be used in and adjacent to the first wall. Mention was also made of nuclear heating and activation effects in the wall, and their effect on its strength. Since NEPTUNE is primarily intended as a plasma modelling tool, this material is neglected here except insofar as it has implications for NEPTUNE interfaces. Similarly ZV has as high priority, a capability to do magnetic equilibrium calculations so as to be able to calculate electromagnetic stresses in the wall. CJ emphasised that engineering design at UKAEA makes heavy use of the ANSYSTM software for multiphysics calculations, and that since plasma effects including neutral beams and fast ions are not part of any standard finite element toolkit, particular consideration should be given to interfacing NEPTUNE to ANSYS macros and ACT (Python scripting for ANSYS). CJ stated that ANSYS had been implementing additional physics very slowly, on a timescale of decades, and there is anyway not a satisfactory licence model for using ANSYS on HPC, nor any to be expected soon.

Although the SMARDDA software takes as input general surface triangulations, which may therefore represent arbitrary surface topologies, SMARDDA-PFC calculates power deposition on the basis of a simple, empirical physical model of transport from the outer midplane. The errors in the model are frequently unquantifiable, given an absence of relevant experimental data, notably when 'small' limiters are proposed, where 'small' implies that many lines of magnetic field make several mid-plane passes before interacting with the limiter(s). The NEPTUNE software should provide a better physical model to enable (1) calculation of the midplane profile of power 'deposition' and associated parameters such as the e-folding length λ_q , when an exponential is fitted, and (2) an assessment of the accuracy of the whole SMARDDA surrogate model in a wide range of existing and novel configurations.

2 Overall Capabilities

CJ has been led to dream of a

Whole-system full-physics digital twin available for real time in-silicon simulation and experimentation

but recognised that a more immediately realisable prospect was integration of plasma software into ANSYS WorkbenchTM.

1. Simulations must be of known, improved accuracy, so that the performance of the built designs can be enhanced without sacrificing safety. Thus the code and data structure should support mesh convergence studies, and be 'physics aware' as described in Section 3.
2. The power load should be easily transferable to ANSYS for stress, heat transfer and other multi-physics analysis.
3. The software should be easy to couple to other physics packages produced within the fusion community such as RACLETTE and ERO for erosion calculations, LOCUST and SMARDDA-PFC for power deposition by particles, and neutronics software such as MCNP.
4. The software should be designed to support easy production of statistics from ensemble calculations, even when costs limit the size of the ensemble. The ensembles should be able to include different model selections as well as different parameter choices.
5. The software should be easy to use on HPC, eg. through cloud-based services, where it should be resilient to network bottlenecks. Its performance should scale well and not depend on the OS used.
6. The software should be able to exploit GPGPUs.
7. The software should use well-defined standard, open formats for both input and output of data.
8. The code should be easy to extend, without writing new code. Thus a user should be able to extend (at least virtually) a data structure to include a new parameter, and add a new physical effect.
9. The following aspects of the software should be open:
 - (a) data structure specification and documentation
 - (b) code and documentation
 - (c) test cases and their documentation
 - (d) all results and their documentation

3 Physics Model

The software should be automatically 'physics aware', ie. it should be able to

1. test the physical assumptions and orderings used, perhaps by calculations at randomly placed points, and when this is impossible, to produce output to enable an independent person to check against a separate code.
2. intelligent enough to switch to a simpler assumption when appropriate or when instructed to do so, for example by using classical transport coefficients when they apply.
3. test the level of detail used. For example, the code (1) could carry out an ensemble calculation using a simplified model of radiative loss, and then check the results in a much shorter time against a fuller model using a sampling technique, (2) be able to identify automatically that a fully 3-D field calculation with say 12 or 18 discrete TF coils has produced a toroidally axisymmetric field in the vacuum vessel.
4. simulate transient behaviour, being able to determine an initial approximate quasi-static solution, then depending on the length of the simulation relative to physical timescales of interest, perform either an implicit or an explicit calculation.
5. in the case that pseudo-time is used, eg. in an implicit calculation, distinguish clearly from physical time.
6. handle incompatible timescales for bulk and local behaviour, eg. account for particle effects on overall flow.

4 Physics Capabilities

The software should be able to

1. compute the total power load on solid walls, due to plasma, fast ions and neutral beams
2. compute the production of impurity species by first wall melt and evaporation.
3. compute heat conduction in first wall coatings in contact with the plasma.
4. perform high frequency electrical and magnetic analysis, accounting for skin effects

5 Geometry

1. The software should be able to account for the effect of changes to geometry caused by radiation swelling, erosion and deposition due to plasma interaction, and corrosion.
2. The code should allow periodic toroidal boundary conditions.
3. The software should be able to handle 0-D, 1-D, 2-D and 3-D representations of the same plasma pulse, transferring between the different representations.
4. A related example concerns the recognition of field axisymmetry as in Section 3.
5. Concavities in the surface, even sharp corners capable of causing singularities in the magnetic and stress fields, should be treatable by the software.

References