PRISM Proposal: Enhancing Configuration and Use of FEM Codes (Phase II Update)

Jeremy Cohen, Department of Computing, Imperial College London – 3-month project extension

Overview The PRISM platform represents a set of groups developing open source software and tools for undertaking finite element analysis in a wide range of scientific fields. Applications such as Nektar++, Fluidity and PyFR present different interfaces to end users or developers but as with almost all advanced scientific applications, they require the setting of a number of configuration parameters to define the task to be undertaken by their users. In the case of complex codes, there can be large numbers of parameters that have a variety of constraints that affect what values can be selected under different usage scenarios or when running on different types of resources. There can also be challenges in managing the full pipeline of processes required to get from an initial geometry to defining and running a job. This project is a 3-month extension of the original 3-months of project funding that will continue to build on the work undertaken and key achievements of the initial project. This work will also continue to build on the TemPSS (Templates and Profiles for Scientific Software) and Nekkloud tools, initially built as part of the EPSRC-funded libhpc (EP/I030239/1) and libhpc Stage II (EP/K038788/1) projects.

The first stage of this project developed new techniques to handle the management of constraints within the TemPSS tool through the use of an embedded constraint satisfaction solver. Meshing support for the latest version of the NekMesh tool was also added to the Nekkloud web application for running Nektar++ jobs. As part of this work it became clear that embedding capabilities to visualise meshes within Nekkloud is now an important requirement. This is also something that is required to support post processing (e.g. video generation) for a wider range of mesh structures than can currently be handled. There is also a further requirement to significantly refactor the Nekkloud application to simplify deployment at remote sites, an area where there is now increased interest. This refactoring will also support the integration of third-party tooling such as ParaViewWeb which is one of the possible routes to enable in-application visualisation of meshes.

Project objectives The core objectives of this work are as follows:

- **Increasing Nekkloud use** via integration of modern build/packaging tools to ease remote deployment
- **Enhancing visualisation capabilities within Nekkloud** to enable mesh visualisation/manipulation
- **Simplify addition of new TemPSS templates** with UI updates that allow upload of template definitions
- **Continue to promote collaboration/sharing** within PRISM using techniques developed in this work
- **Publish first stage project outputs** covering the constraint solving capabilities added to TemPSS

This second phase of the project will continue to **prototype** new features within the Nekkloud and TemPSS tools with a particular focus on improving **ease of deployment** for Nekkloud at remote sites, something that there has been interested in as a result of dissemination from the first stage of the project. This requires extensive **refactoring** of the application and integration of the latest build and packaging tools for web applications. This will, in-turn, support the integration of libraries and tools to support **mesh visualisation** directly within Nekkloud, addressing a current problem that requires users to undertake manual visualisation of meshes generated within the meshing console, outside of Nekkloud, in order to check their correctness. This reduces the effectiveness and usability of the Nekkloud application. A publication covering the constraint management features developed in phase 1 will also be prepared during this phase of work.

Alignment with PRISM strategy

Retention of key staff: This project will provide further short-term bridging of Dr Jeremy Cohen who has worked extensively with members of the PRISM platform for some time. He has significant knowledge and experience, developed over several years, of processes for simplifying access to complex scientific codes and computational infrastructure and has been involved in the development of a range of tools and services.

Longer-term research: Phase 2 of this project builds on outputs of previous EPSRC-related projects and continues work from phase 1. It is considered to be of key importance in enabling additional capabilities that can bring codes from the PRISM platform to a wider community of domain scientists and researchers.

Collaboration between PRISM members and groups: The aim of this project is to develop and demonstrate techniques and approaches that are extensible and transferrable across the PRISM platform. As a result the project will look to apply outputs to other codes in the PRISM domain where this is feasible.

Brief work plan This work will build on outputs of the first stage project, focusing on enhancing the tools and services prototyped and improving their usability and accessibility. Key tasks/milestones are as follows:

- Extensive refactoring of the Nekkloud code and adoption of frontend build/packaging tools to ease deployment and maintenance.
- Integration of visualisation capabilities into the meshing console enabling an end-to-end job pipeline.
- Modify the TemPSS user interface to allow upload of template and constraint definition files. This will enable easy addition of new templates to remote TemPSS deployments without rebuilding code.
- Prepare a publication covering the constraint solving features added to TemPSS