PRAgMaTIc impact plan

Proposers:	Gerard Gorman, Michael Lange, Patrick Farrell
Candidate:	Georgios Rokos (PhD student supervised by Paul Kelly and Gerard Gorman - submitting thesis in August 2014)
Resources requested:	*Underwrite* 6 months salary to start on the 1st of September 2014. A derivative of this proposal will also be submitted to the Imperial College EPSRC impact award fund (provisionally targeting a submission for the October review board). If that application is successful the candidate will immediately switch to the EPSRC impact award.

PRAgMaTIc (Parallel anisotRopic Adaptive Mesh ToolkIt) provides 2D/3D anisotropic mesh adaptation for meshes of simplexes. The target application are finite element and finite volume application codes although it can also be used as a lossy compression algorithm for 2 and 3D data (*e.g.* image compression). PRAgMaTIc takes as its input a mesh and a vertex-centered metric tensor field that encodes desired mesh element size anisotropically. The toolkit is written in C++ and has OpenMP and MPI parallel support.

To date PRAgMaTIc has primarily been a research project funded by Fujitsu Laboratories of Europe Ltd and an EPSRC studentship, to investigate methods for optimizing the scaling irregular computation and redesign mesh adaptation for the multi-/many-core era. In contrast, the project proposed here focuses on code hardening and integration into existing simulators so that PRAgMaTIc can be used robustly in application codes, thereby delivering impact to the end users of the application codes. The proposed work packages are as follows:

1. Integrate 2D and 3D PRAgMaTIc with PETSc/DMPlex. This will facilitate PRAgMaTIc being used by application codes that use PETSc, including the open source simulators supported by the EPSRC PRISM platform Grant: Fluidity, Nektar++ and Firedrake. This work package will engage the development teams of these application codes to support integration of mesh adaption. The diversity of these codes will help ensure that PRAgMaTIc's application user interface, and extensions to PETSc/DMPlex, are sufficiently general to support a wide range of other simulators. Dr Matthew Knepley, one of the key PETSc developers, is committed to support the development and integration of features into mainline PETSc.
2. Integrate 2D and 3D PRAgMaTIc into FEniCS. This will facilitate PRAgMaTIc being used by application codes that use FEniCS and DOLFIN. There is a significant technology overlap between Firedrake and FEniCS, the latter being more mature and having a larger community. Making PRAgMaTIc available to the FEniCS community will accelerate its adoption into the wider community, thereby increasing its impact and accelerating its further development. Some work in integrating 2D PRAgMaTIc has already been completed and is actively being used in Kristian Jensen's work on shape optimisation. For this work additional travel and subsistence support is requested to allow Georgios to work with Patrick Farrell at Oxford who has an indepth knowledge of FEniCS and specific test cases that can be used for verification and demonstration.
3. Implement a generic toolkit for mesh-to-mesh interpolation. The numerical details of mesh-to-mesh interpolation are dependent on the numerical discretisation used, and thus the specific application code. However, there are generic geometric operators required to facilitate application codes to implement the required interpolation operator. In the first instance this will involve lifting the Fortran 90 code written by Dr Patrick Farrell for Fluidity, along with a minimum code dependency set, to create a standalone library that could be called from 3rd party software. Consistent with the

rest of our strategy, we will make the library PETSc ready (i.e. will add a --download-supermesh to PETSc) as that will maximize visability and impact. Specific functionality that will be made available includes:

   a. Locate elements in source mesh that contain nodes in target mesh. The container element may be on a remote process. Application codes will typically use this information to perform colocated interpolation using finite element shape functions. In addition, such a mapping is also required by PETSc to facilitate geometric multigrid preconditioning via coarse meshes created by PRAgMaTIc.
   b. For a given element on the target mesh, calculate the *supermesh* on the source mesh. This is a critical building block to allow the application code to implement conservative interpolation.

   For this work package we are also requesting additional travel and subsistence support for Georgios to work with Patrick Farrell.

4. Refactor PRAgMaTIc software for greater reuse of source code between 2D and 3D mesh adaptation. The 2D and 3D mesh adaptation implementations diverged during the course of Georgios Rokos's PhD as research into optimising of irregular computation was carried out primarily for 2D. This work-package will allow the same optimised OpenMP and MPI specific code to be reused for both 2D and 3D mesh adaptation.
5. Write developers and users manual for PRAgMaTIc. This will include tutorial examples based on PRAgMaTIc s test suite. A resulting release of the PRAgMaTIc package may then be hosted on the PETSc package server and made available via PETSc's build system.