Technical Report (publishable)

[approx 5-10 pages, although longer is permissible if it is appropriate]. This should provide a summary of the technical work completed on the project, including performance results and other relevant technical information. The technical report will be published on the ARCHER web site and should read as a standalone document hence you are welcome to replicate text from elsewhere in this document. Please begin the document with an abstract including the key scientific aims of the project and corresponding achievements.

This section may be supplied as a separate PDF. Please ensure the author list is complete.

Please make sure this has an abstract and conclusion and appropriately acknowledges ARCHER and the eCSE programme:

"This work was funded under the embedded CSE programme of the ARCHER UK National Supercomputing Service (http://www.archer.ac.uk)"

Delivering a step-change in performance and functionality to the Fluidity shallow water solver through code generation

C. T. Jacobs, M. D. Piggott, D. A. Ham

1. Abstract

Fluidity (fluidity-project.org) is a fully-featured, open source, computational fluid dynamics (CFD) framework. It comprises several advanced numerical models based on the finite element method as well as a number of novel numerical features (e.g. mesh adaptivity) making it suitable for multi-scale simulations (Piggott et al., 2008). It is largely unique in its abilities to also solve large-scale geophysical/oceanographic problems. Key examples include marine renewable energy, tsunami simulation and inundation, and palaeo-tidal simulations for hydrocarbon exploration (Martin-Short et al., 2015; Mitchell et al., 2010; Mitchell et al., 2011; Oishi et al., 2013; Shaw et al., 2008).

The current Fluidity codebase comprises hand-written Fortran code to perform the finite element discretisation. Not only is this hand-written code potentially sub-optimal, it presents issues regarding its maintainability and longevity; should one want to run Fluidity on a newer hardware architecture more suited to larger scale problems in the future, then the entire codebase may have to be re-written. Furthermore, the need for numerical modellers to not only be experts in their field of science, but also be well-versed in parallel programming and code optimisation, is unsustainable in the long-term.

This eCSE project delivers a step-change in the performance and functionality of the shallow water model within Fluidity, accomplished by using the Firedrake (firedrakeproject.org) framework for the automated solution of partial differential equations using code generation techniques (Rathgeber et al., Submitted). A key aim is to remove Fluidity's existing hand-written Fortran finite

element discretisation code and instead generate it automatically from a higher-level model description (written in a language known as the Unified Form Language, UFL (Alnaes et al., 2014)), thereby hiding complexity through layers of abstraction. This allows the users of the resulting models to focus on the problem specification and the end results of simulations. The Firedrake project achieves all of this in a performance-portable manner using the PyOP2 framework (Markall et al., 2013; Rathgeber et al., 2012) to target and optimise the automatically-generated code for a desired hardware architecture. Moreover, in contrast to traditional, hand-written models, such as Fluidity, the use of code generation techniques has been shown to deliver significantly enhanced performance, as well as improved code maintainability (e.g. Maddison and Farrell (2014); Olgaard and Wells (2010)).

2. Project Outcomes

2.1 Re-engineering

The first phase of the project involved building upon some recently completed proof-of-concept work (Jacobs and Piggott, 2015) on the use of Firedrake to solve idealised shallow water (SW) problems, to replicate Fluidity's current SW solver using code generation techniques. This model comprises a collection of files, mostly written in the high-level Unified Form Language (Alnaes et al., 2014) which describes the governing equations in their 'weak/variational form'.

An initial step was to extend the proof-of-concept code to solve both the steady (i.e. non time dependent) and unsteady (i.e. time dependent) forms of the shallow water equations. This was accomplished by adding a new option to the Fluidity simulation configuration files to enable/disable the time derivative term in the shallow water equations. These extensions were then verified and validated to demonstrate the model's correctness; the results of this testing process are detailed in the next section.

Clearly the re-engineering of every model in the whole Fluidity codebase is a major task. Therefore, during the transition, users are given the option to continue using the old legacy code or switch to the new automatically-generated code developed with Firedrake which will provide performance portability on ARCHER and its successors. This is accomplished by a pre-processing program developed during this eCSE to convert 'old style' Fluidity simulation configuration files to a 'new style' one expected by the new re-engineered model.

2.2 Testing & benchmarking

A rigorous testing campaign incorporating a range of unit, regression, analytical, and manufactured solution tests, as well as real-world case studies, was undertaken to ensure the correctness of the new re-engineered model. The figures below illustrate two such test cases, used to verify and validate the code.

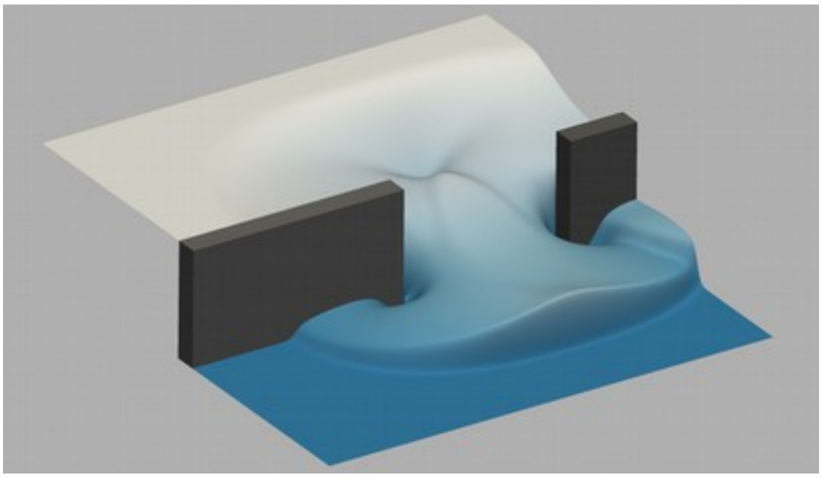

Figure: A visualisation of a numerical simulation of flooding following a dam breach. The setup follows that of Liang et al. (2008) and considers a high-level of water that is initially held back by a dam wall (shaded grey). A 75 metre-wide breach in the wall is then considered, through which water rushes into the lower section to form a tidal bore wave. Swirling vortices are visible near the corners of the breach. The simulation was performed on Archer to validate the numerical model.

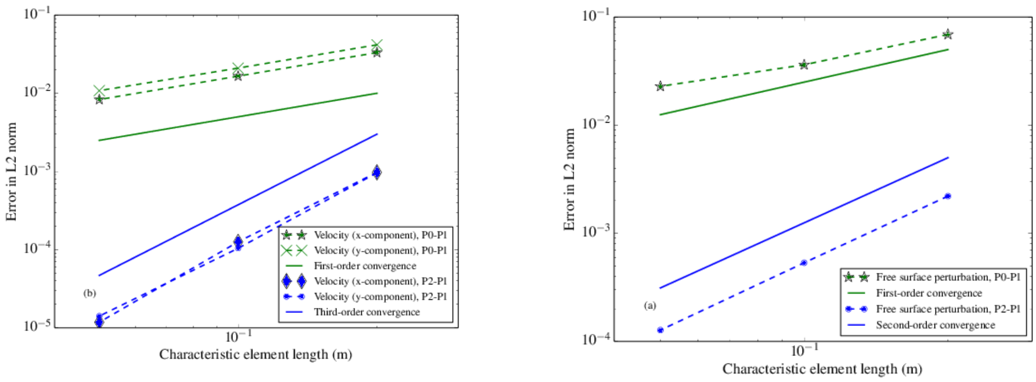

Figure: The results from the MMS (Roache, 2002) convergence analysis. Both the P2-P1 and P0-P1 element pairs were considered to check the correctness of both the CG and DG finite element methods implemented in the re-engineered code. The left-hand figure demonstrates that the velocity in the P2-P1 and P0-P1 simulations converge at third-order and first-order as expected. The right-hand figure shows the convergence results for the free surface field. For the P2-P1 simulation this is second-order as expected for a free surface field represented by P1 (linear) polynomial basis functions. The first-order convergence in the P0-P1 field is likely a result of the coupling with the lower-order velocity field and the use of a first-order upwinding scheme for advection across discontinuous element facets.

Benchmarking of the old and new solvers on ARCHER followed from performance data collected from these test cases. Some strong scaling results are shown in the figures below. The assembly stage for the re-engineered code is significantly faster than the legacy code (approximately half an order of magnitude faster) and scales well. However, the solver costs are higher as a result of the difference in the solution method used. Solving the fully-coupled system in the re-engineered code is very costly and requires a different PETSc (Balay et al., 2014) solver configuration (GMRES (Saad and Schultz, 1986) with a more complex Fieldsplit preconditioner (Brown et al., 2012), compared to GMRES with SOR), although it may be more robust than the projection method used in the legacy code.

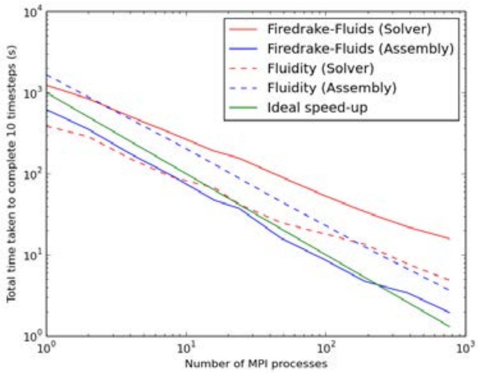

Figure: Run-time against the number of MPI processes on ARCHER, for the 2D dam break problem. This considered the P2-P1 element pair with approximately 3 million mesh vertices.

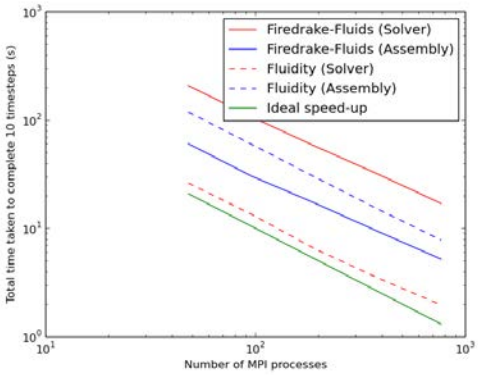

Figure: Run-time against the number of MPI processes on ARCHER, for the flow past a square problem. This considered the P2-P1 element pair with approximately 1 million mesh vertices.

2.3 New Functionality

The fact that the re-engineered code is based upon automatically-generated code opens up huge possibilities for bringing important new functionality to Fluidity's user community, enabling new science and engineering to be conducted. Firedrake interfaces with the FIAT finite element (FE) tabulator from the FEniCS project (Kirby, 2004) to provide practically unlimited orders of basis functions to discretise the solution fields in space. The legacy Fluidity code is currently limited to a maximum order of 3, whereas users can now simply modify the “polynomial order” option in a simulation's configuration file in order to use higher orders, enabling more accurate simulations and easier usability from a model developer's perspective.

Three element pairs (P0-P1, P1DG-P2 and P2-P1) were evaluated to provide guidelines for suitable discretisations for different classes of problem in order to achieve optimal computational performance on ARCHER, which will further help to enable previously untenable simulations. A single tidal turbine in a steady channel flow was considered, as well as the spread of a Gaussian 'bump' in the free surface field to approximate a tsunami wave.

It was found that the P0-P1 element pair, although computationally inexpensive, resulted in significant numerical diffusion of the 'tsunami' wave due to the low order of the basis function used to represent the velocity field. The P1DG-P2 pair also featured considerably reduced wave velocity, compared to the P2-P1 simulation. This is likely a result of excessive numerical diffusion from the upwinding advection scheme featured in the discontinuous Galerkin finite element method's implementation. More advanced schemes such as slope limiters could potentially reduce this numerical diffusion and maintain a sharper wave front; however, such schemes are not expressible in UFL and require implementation as a lower-level C kernel which can be passed directly to PyOP2. In the tidal turbine simulations, the P1DG-P2 and P2-P1 element pairs gave comparable wake lengths and behaviour, but the effect of the turbine on the flow was almost unnoticeable with a P0-P1 discretisation, again as a result of the low order of basis function used. For production simulations, such as the one shown in the figure below, the P2-P1 was used throughout as a result of its robustness and ability to maintain a relatively high order of accuracy.

The other piece of new functionality brought about by the use of Firedrake is the integration of the libadjoint library (Farrell et al., 2013) to automatically 'differentiate' the re-engineered shallow water solver. Developing adjoints to models is very challenging, but code generation has opened up new possibilities to do this in a rapid and sustainable manner. Preliminary simulations involving flow past tidal turbine arrays (see figure below) have shown how varying the position of each turbine can affect the total amount of power generated due to wake effects and reduced drag. However, man-made 'naïve' configurations are not necessarily optimal. Adjoint modelling has the potential to optimise the positions of the turbines in order to maximise the amount of power generated (Funke et al., 2014).

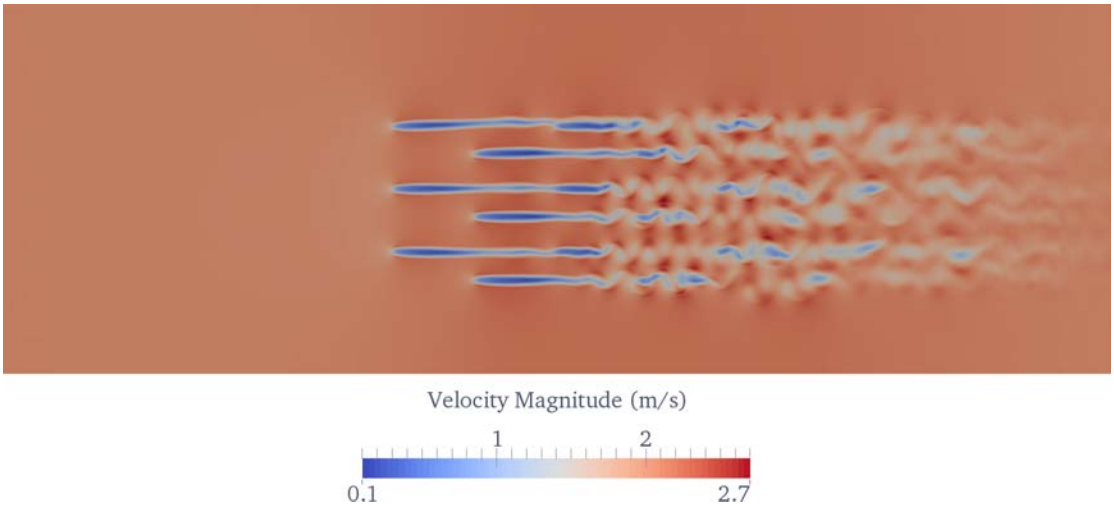

Figure: Simulation of flow past a set of 15 tidal turbines (each of size 5 x 20 x 50 m) in a channel with a peak flow velocity of 2 m/s. This simulation was performed on ARCHER using a mesh of ~100,000 vertices distributed over 48 MPI processes. The magnitude of the velocity at t = 1,116 s is shown in the entire 3 km x 1 km domain. A Smagorinsky LES model was used to parameterise the wake turbulence. Staggering the turbines in this way produces significantly greater power output compared to simple grid-like layouts or increasing the spacing between each turbine.

2.4 Training, user engagement and impact

A new set of user documentation has been written for the re-engineered code, and is now automatically built and hosted online at: http://firedrake-fluids.readthedocs.org. User engagement and impact will follow from the extensive (and growing) academic and industrial user community being able to solve larger and more challenging problems with the re-engineered code, thus allowing higher solution fidelity (less parameterisation) and enhanced confidence in the study of important and timely issues related to energy security and natural hazards. Furthermore, the new functionality in the re-engineered code allows this user community to take advantage of the adjoint technology that has already been demonstrated for the optimisation of tidal turbine arrays in the OpenTidalFarm package (Funke et al., 2014), and to also consider optimisation, sensitivity, and uncertainty questions in new application areas.

3. Conclusion

Through the insertion of code generation techniques into the Fluidity CFD framework, this eCSE project has successfully delivered a significant performance improvement in the finite element assembly stage of Fluidity's shallow water model. With automatic code optimisations, matrix and form caching techniques, and the efficient execution of the code using the PyOP2 library (Markall et al., 2013; Rathgeber et al., 2012), the assembly operations are almost an order of magnitude faster than the legacy, hand-written Fortran codebase. Furthermore, this project has prepared Fluidity for running the same models on modern state-of-the-art high-performance hardware architectures, such as the new Intel Xeon Phi coprocessors, without the need for model developers to re-write any of their existing code.

While the run-time required to assemble the system of model equations has been reduced, the run-time spent in the linear solvers is not currently comparable with the legacy code due to the differences in the overall solution methods used;

the legacy code uses a relatively inexpensive projection method, compared to the Firedrake-based code which solves the fully-coupled matrix system with complex fieldsplit preconditioners. A similar projection method (known as IPCS (Goda, 1979)) was also implemented within the new Firedrake-based code; here the solver times were comparable to those of the legacy code, but IPCS was found to not provide particularly robust solutions.

Several new pieces of functionality have been introduced that will enable the expansion of Fluidity's existing userbase and allow users to tackle large, interesting scientific problems. These include the integration of the FIAT library for generating basis functions of arbitrary order to permit higher solution accuracy, and also the integration of the Firedrake-Adjoint library (Farrell et al., 2013; Funke et al., 2014) which has a wide range of possible applications. The preliminary work on tidal turbine modelling performed during this eCSE project is currently being extended to investigate the validity of steady-state adjoint computations using non-physically large viscosity values (performed using the optimisation functionality in the OpenTidalFarm package (Funke et al., 2014)), and comparing the results with lower viscosity, transient runs which parameterise the turbine wakes with the LES turbulence model.

4. Acknowledgements

This work was funded under the embedded CSE programme (eCSE 03-7) of the ARCHER UK National Supercomputing Service (http://www.archer.ac.uk).

References

M. S. Alnaes, A. Logg, K. B. Olgaard, M. E. Rognes, and G. N. Wells (2014). Unified Form Language: A domain-specific language for weak formulations of partial differential equations, *ACM Transactions on Mathematical Software*, 40.

S. Balay, Abhyankar, S., Adams, M., Brown, J., Brune, P., Buschelman, K., Eijkhout, V., Gropp, W., Kaushik, D., Knepley, M., Curfman McInnes, L., Rupp, K., Smith, B., and Zhang, H. (2014). PETSc Users Manual, Tech. Rep. Revision 3.5, Argonne National Laboratory.

J. Brown, M. G. Knepley, D. A. May, L. C. McInnes, B. F. Smith (2012). Composable Linear Solvers for Multiphysics, In *Proceeedings of the 11th International Symposium on Parallel and Distributed Computing (ISPDC 2012)*, 55–62, doi:10.1109/ISPDC.2012.16.

P. E. Farrell, D. A. Ham, S. W. Funke, M. E. Rognes (2013). Automated derivation of the adjoint of high-level transient finite element programs. *SIAM Journal on Scientific Computing* 35:C369-C393. Doi:10.1137/120873558

S. W. Funke, P. E. Farrell, M. D. Piggott (2014). Tidal turbine array optimisation using the adjoint approach. *Renewable Energy* 63, 658-673. doi:10.1016/j.renene.2013.09.031

K. Goda (1979). A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows. *Journal of Computational Physics* 30, pages 76-95.

C. T. Jacobs, M. D. Piggott (2015). Firedrake-Fluids v0.1: Numerical modelling of shallow water flows using a performance-portable automated solution framework. *Geoscientific Model Development* 8, 533-547, doi:10.5194/gmd-8-533-2015

R. C. Kirby (2004). Algorithm 839: FIAT, a New Paradigm for Computing Finite Element Basis Functions. *ACM Transactions on Mathematical Software*, 30, 502–516, doi:10.1145/1039813.1039820

S.-J. Liang, J.-H. Tang, and M.-S. Wu (2008). Solution of shallow-water equations using least-squares finite-element method, *Acta Mechanica Sinica*, 24, 523–532, doi:10.1007/s10409-008-0151-4

F. Luporini, A. Varbanescu, F. Rathgeber, G.-T. Bercea, J. Ramanujam, D. A. Ham, P. H. J. Kelly (2015). Cross-Loop Optimization of Arithmetic Intensity for Finite Element Local Assembly. *ACM Transactions on Architecture and Code Optimization (TACO)* 11(4). doi:10.1145/2687415

J. R. Maddison, P. E. Farrell (2014). Rapid development and adjoining of transient finite element models. *Computer Methods in Applied Mechanics and Engineering* 276:95-121. doi:10.1016/j.cma.2014.03.010

G. R. Markall, F. Rathgeber, L. Mitchell, N. Loriant, C. Bertolli, D. A. Ham, P. H. J. Kelly (2013). Performance-portable finite element assembly using PyOP2 and FEniCS. *Supercomputing*. Springer, 279–289. doi:10.1007/978-3-642-38750-0_21

R. Martin-Short, J. Hill, S.C. Kramer, A. Avdis, P.A. Allison, M.D. Piggott (2015). Tidal resource extraction in the Pentland Firth, UK: Potential impacts on flow regime and sediment transport in the Inner Sound of Stroma. *Renewable Energy* 76:596-607. doi:10.1016/j.renene.2014.11.079

A. J. Mitchell, P. A. Allison, G. J. Gorman, M. D. Piggott, C. C. Pain (2011). Tidal circulation in an ancient epicontinental sea: The Early Jurassic Laurasian Seaway. *Geology* 39, 207-210. doi:10.1130/G31496.1

A. J. Mitchell, P. A. Allison, M. D. Piggott, G. J. Gorman, C. C. Pain, G. J. Hampson (2010). Numerical modelling of tsunami propagation with implications for sedimentation in ancient epicontinental seas. *Sedimentary Geology* 228, 81-97. doi:10.1016/j.sedgeo.2010.03.008

Y. Oishi, M. D. Piggott, T. Maeda, S. C. Kramer, G. S. Collins, H. Tsushima, T. Furumura (2013). Three-dimensional tsunami propagation simulations using an unstructured mesh finite element model. *Journal of Geophysical Research: Solid Earth* 118, 2998-3018. doi:10.1002/jgrb.50225

K. B. Olgaard, G. N. Wells (2010). Optimizations for quadrature representations of finite element tensors through automated code generation. ACM Transactions on Mathematical Software (TOMS) 37(1). doi:10.1145/1644001.1644009

M. D. Piggott, G. J. Gorman, C. C. Pain, P. A. Allison, A. S. Candy, B. T. Martin, M. R. Wells (2008). A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes, *International Journal for Numerical Methods in*
Fluids 56, 1003-1015. doi:10.1002/fld.1663

F. Rathgeber, G. R. Markall, L. Mitchell, N. Loriant, D. A. Ham, C. Bertolli, P. H. J. Kelly (2012). PyOP2: A High-Level Framework for Performance-Portable Simulations on Unstructured Meshes.
Supercomputing Companion, 1116–1123. doi:10.1109/SC.Companion.2012.134

F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G.-T. Bercea, G. R. Markall, P. H. J. Kelly (Submitted). Firedrake: automating the finite element method by composing abstractions. Submitted to ACM TOMS.

P. J. Roache (2002). Code Verification by the Method of Manufactured Solutions. *Journal of Fluids Engineering*, 124, 4–10, doi:10.1115/1.1436090

Y. Saad, M. H. Schultz (1986). GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM Journal on Scientific and Statistical Computing, 7, 856–869, doi:10.1137/0907058

B. Shaw, N. N. Ambraseys, P. C. England, M. A. Floyd, G. J. Gorman, T. F. G. Higham, J. A. Jackson, J.-M. Nocquet, C. C. Pain, M. D. Piggott (2008). Eastern Mediterranean tectonics and tsunami hazard inferred from the AD 365 earthquake, *Nature Geoscience* 1, 268-276. doi:10.1038/ngeo151