Response to the reviewers

January 22, 2015

Reference: gmd-2014-155

The changes made to the manuscript are highlighted in the file `diff.pdf` in the supplement to comment C3049: http://www.geosci-model-dev-discuss.net/7/C3049/2015/gmdd-7-C3049-2015-supplement.zip

All page numbers and line numbers mentioned in the response refer to those in `diff.pdf`.

Reviewer 1: M. Rognes

General Comment 1

"The manuscript strongly emphasizes the performance benefits of Firedrake in the introduction (page 5702, L5-16). It is clear that the Firedrake strategy shows promise, but large-scale performance and performance portability has not yet been demonstrated. Therefore, the claims presented in this manuscript ("Firedrake provides enhanced performance benefits...", "Firedrake is at least as fast, if not faster than ...") seems lacking of nuance and too strongly biased. I would recommmend that the authors seek to refine this presentation."

Large-scale performance-portability with Firedrake has indeed not yet been demonstrated, and we are not yet in a position to be able to show this for the types of problems we present in this paper due to current limitations in Firedrake/PyOP2 (e.g. the assembly and solution of non-linear problems cannot currently be performed on GPUs, see page 14 lines 15–17). We have therefore removed the introductory paragraphs:

"Recent application of PyOP2's code optimisation strategies has demonstrated up to a factor 4 speed-up compared to running FEniCS-generated code (Luporini et al., Submitted). Furthermore, for a suite of benchmark problems (including Cahn-Hilliard, advection-diffusion and Poisson equation-based problems), Firedrake is at least as fast, if not faster, than the FEniCS framework (Rathgeber, Submitted)." (page 4, lines 12–17)

"Essentially, Firedrake provides the same high-level problem solving interface, with enhanced performance benefits." (page 4, lines 8–9)

and moved any sentences regarding performance-portability to the 'future work'/roadmap section (page 53, lines 6–10). We have also removed "performance-portable" in the manuscript's title, and have updated the description of Firedrake itself (page 3 line 24 – page 4 line 3). This now makes reference to a pre-print of a paper (Rathgeber *et al.*, Submitted) about the Firedrake library, which has recently been submitted to TOMS.

General Comment 2

"The use of block preconditioning is an active research topic and one of substantial interest to the research field. Please include more detail regarding the fieldsplit preconditioners, their set-up, choice of parameter values et cetera. Also, please include at least one example documenting the performance of the iterative solvers (including iteration numbers versus system size and parallel scaling)."

We have added further information on the fieldsplit preconditioner in Section 3.3 (page 20, lines 3–19), including the method used to solve for the action of the inner blocks of the Schur complement factorisation. Parallel scaling performance of the iterative solver and preconditioner are now given for the 2D dam break problem in Section 4.2.1 (page 35 line 1 – page 37 line 11) for two different fieldsplit setups, as well as details on the number of iterations typically taken.

General Comment 3

"I recommend that the authors include a subsection in Section 3 describing the Smagorinsky LES model and its implementation in Firedrake fluids, thus moving and extending the description in Section 4.5."

We felt that the description of the Smagorinsky LES model would be better placed in Section 2 (the "Model equations" section), and have therefore created a new sub-section entitled "Turbulence modelling" (Section 2.4, page 9 line 10 – page 10 line 3) which contains this description. The details of the model's implementation in Firedrake-Fluids can be found in Section 3.4 (page 24 lines 7–25).

General Comment 4

"The manuscript strongly emphasizes the performance portability of the underlying framework, however, there is no mention of the performance of the Firedrake-fluid model nor how and on which architectures the presented test cases are run. The manuscript would benefit substantially from remedying this mismatch. Please include - information on the architecture and run-times for the numerical experiments - one experiment that demonstrates performance portability across at least two backends."

We have included information on the architectures used to run the various numerical simulations presented throughout the paper, along with the run-times (e.g. page 30, lines 1–10, page 38 lines 18–20, page 43 lines 3–5, page 47 lines 7–9). Due to limitations in Firedrake and PyOP2, we are currently unable to run Firedrake-Fluids on more than one hardware architecture. While Firedrake/PyOP2 does have limited support for OpenCL and CUDA backends, they do not currently support the assembly or solution of *non-linear* problems on GPUs. In addition to the standard pure MPI runs, we are experimenting with a 'hybrid' approach comprising MPI and OpenMP on CPUs, but this is still a work-in-progress and are not able to demonstrate performance-portability at this point. This is something that will be investigated in greater detail in the future, and have adjusted the text accordingly throughout the manuscript (please see the response to General Comment 1, and page 54 lines 17–19 and page 55 lines 11–14).

Specific Comment 1

"P5702, L3-4: This example (caching of matrices) is not appropriate as an argument for the advantages of Firedrake vs FEniCS as both easily support this construction."

We have removed this sentence regarding the caching of matrices (page 4, lines 6–8).

Specific Comment 2

"Pp5704, L6: I assume that u also depends on time in addition to the spatial coordinates (x, y). Please consider specifying."

The velocity $\mathbf{u}$ does indeed depend on the spatial coordinates as well as time, and we have now added t alongside x and y (page 5, line 20). Similarly, we have also highlighted the dependence of h on x, y and t, and the dependence of H on x and y (page 5, lines 21–22).

Specific Comment 3

"Eq (1): Please specify the norm $||.||$"

We have specified which norm we have used here: "The Euclidean norm $||\mathbf{u}||_2 = \sqrt{\mathbf{u} \cdot \mathbf{u}}$ is used here.", on page 7 line 8.

Specific Comment 4

"Eq (1) & (3): Please specify the unknowns (u, h?), initial and boundary conditions."

We have now specified the unknown fields, $\mathbf{u}$ and h (page 7, line 2). The general form of their initial and boundary conditions are defined in a new subsection entitled "Initial and boundary conditions" (Section 2.3, page 8).

Specific Comment 5

"Solving PDEs such as (1) + (3) in Firedrake/FEniCS requires a temporal and spatial discretization of the PDE, as reflected by the code example in Fig 2. Please clarify this in the 1st paragraph of Section 3.1"

We have added the requirement of temporal and spatial discretisation to the first paragraph in Section 3.1 (page 10, lines 7–8).

Specific Comment 6

"The motivation and implications of Footnote 1 are not entirely clear. Please upgrade this footnote and elaborate. In particular, for u in DG0, grad u = 0 and thus T = 0?"

We have updated this footnote on page 7 to explain how the stress tensor is treated when using DG, and that the form of the stress tensor is currently restricted to keep the implementation as simple as possible, not because of a restriction in UFL.

Specific Comment 7

"P5705: Footnote 2: Please indicate the precise version of this "modified FFC" used and preferably a reference for the sake of reproducibility."

Page 14, footnote 2: We have added a link to the modified FFC project's Bitbucket repository, and have specified the precise revision of the `master` branch used throughout the simulations presented in the manuscript.

Specific Comment 8

"Examining Figure 6b, it seems that the convergence of the P2 velocity field is indeed higher than 2nd order. Please comment."

The velocity field's convergence is indeed higher than second-order, and is in fact closer to third-order (at least $O(2.87)$) as expected from a P2 function space. We have therefore updated Figure 6 (page 29) to show a third-order convergence line next to the data points, and have updated the paragraph discussing the order of convergence (page 28, lines 2–5). On page 28, lines 13–23, we have commented on the likely reason why the convergence of the P1 free surface field in the P0-P1 is only first-order.

Specific Comment 9

"The implicit Euler scheme is expected to yield first order convergence in time; the second order convergence(s) observed thus indicates that the spatial error dominates. Please comment."

On page 28, lines 5–12, we have now made it clearer how the time-scale and spatial length-scales are related to the leading error in the MMS simulations, and that for the particular choice of Δt and Δx considered, the error in Δx dominates.

Technical Comment 1

"P5701, L10-11: it is unclear what the partial sentence "..., rather than by hand" refers back to"

We have clarified this sentence (on page 3 line 8) by changing it to "..., rather than the user having to write the low-level code themselves.".

Reviewer 2: A. Dedner

Summary Comment 1

"I think that the introduction should be written a bit more balanced and taking into account additional options for users looking for a fluid dynamics solver. The numerical results should be extended to actually demonstrate the claimed flexibility with regards to different computer architectures and the resulting efficiency of the code."

Due to current limitations in Firedrake and PyOP2, we are unable to run Firedrake-Fluids on more than one hardware architecture. While Firedrake/PyOP2 does have limited support for OpenCL and CUDA backends, they do not currently support the assembly or solution of *nonlinear* problems on GPUs. In addition to the standard pure MPI runs, we are experimenting with a 'hybrid' approach comprising MPI and OpenMP on CPUs, but this is still a work-in-progress and are not able to demonstrate performance-portability at this point. This is something that will be investigated in greater detail in the future, and have adjusted the text accordingly throughout the manuscript (please see the response to General Comment 1 of Reviewer 1, page 54 lines 17–19 and page 55 lines 11–14).

Summary Comment 2

"The results presented do not go into any detail concerning the machines used (are any computation done on a GPU or in parallel?). Also important information like parallel scaleup or other efficiency measures are not reported on at all."

We have included information on the architectures used to run the various numerical simulations presented throughout the paper, along with the run-times (e.g. page 30, lines 1–10, page 38 lines 18–20, page 43 lines 3–5, page 47 lines 7–9). Parallel scaling performance of the iterative solver and preconditioner are now given for the 2D dam break problem in Section 4.2.1 (page 35 line 1 – page 37 line 11) for two different fieldsplit setups, as well as details on the number of iterations typically taken.

Summary Comment 3

"Finally the for me most interesting aspect concerning the flexibility in extending or customizing the code is not addressed. This could include the options for preconditioners used, time integration methods, or the inclusion of the LES model in the code."

We have added details regarding the type/orders of basis function that is currently available to users (page 19 lines 9–13), the different linear solvers and preconditioners that are available through PETSc (page 19 lines 26–27) and the different Fieldsplit setups that have so far been considered (page 37 lines 1–11), and also the LES model (please see response to Detailed Comment 3 below). Currently there is only one time integration method available, and we have now made this clearer in the manuscript (page 18 line 22 – page 19 line 2).

Detailed Comment 1

"Concerning claim (a): in many places in the paper the number of lines of code are compared: a few hundred for the new model compared to many thousands for "handwritten code". In many packages a simple shallow water model as considered here could be easily implemented with a few hundred lines of code (or even less). Packages like DealII, Dune, FreeFem etc. use interfaces to allow users to easily implement their own model problem without much coding required. The number of code lines is from my point of view not the selling point of a UFL bases code generator but the use of the mathematical notation (this is mentioned) and the flexibility to use different and future platforms (mentioned but as already stated not demonstrated)."

We have made it clearer that the benefits of writing fewer lines of code is not unique to automated code generation approaches. We have also made reference to the OpenFOAM, deal.II, Dune and FreeFem++ packages as examples which allow models to also be implemented with relatively few lines of code (page 3 lines 18–21, page 55 lines 5–8).

Detailed Comment 2

"some of the problems with the UFL approach do become apparent in this paper but are not followed up on. It is stated for example that using general diffusion tensors with a discontinuous ansatz space is not yet possible. So if a user wanted to use a better diffusion model, they would be forced to implement quite a few lines of extra code (100 or 1000 perhaps). The authors do not go into detail on why this restriction has to be imposed and if there is some more fundamental problem with the code generation used."

We have updated the footnote on page 7 to explain how the stress tensor is treated when using DG, and why the form of the stress tensor is currently restricted. We have also added a paragraph (at the end of the 'Spatial and temporal discretisation' section, page 19 lines 14–20) discussing an example of where UFL cannot be used and how a user could implement their method in lower-level C code by using a PyOP2 kernel which operates directly on the nodal values in the mesh.

Detailed Comment 3

"An extension is mentioned (an LES model) but how this is implemented within the framework is not covered although this would be very interesting to see."

We have moved the description of the Smagorinsky LES model to a new sub-section entitled "Turbulence modelling" (Section 2.4, page 9) which contains this description. The details of the model's implementation in Firedrake-Fluids can be found in Section 3.4 (page 24 lines 7–25). Please note that the "SUBSCRIPTNB" in this section has been generated by latexdiff, and is not a typographical error in the paper's LaTeX source.

Small Remark 1

"the observed convergence rates are not clear to me since the temporal discretization is only first order. Nevertheless second order convergence is observed. How is the time step and spatial length

scale coupled?"

On page 28, lines 5–11, we have detailed how the time-scale and spatial length-scales are related to the leading error in the MMS simulations, and that for the particular choice of Δt and Δx considered, the error in Δx dominates.

Small Remark 2

"the ordering in terms between equation (4) and (8) should be made consistent."

We have modified equations (8) and (9) (these are now numbered (15) and (16)) to be consistent with the ordering in equation (4) (now equation (11)). Note that the signs in front of the matrices on the RHS are now consistent as well.

Small Remark 3

"Give some specification of the boundary conditions available and which ones are used in the numerical experiments."

We have added a subsection (Section 2.3, pages 8–9) defining the available boundary conditions (Dirichlet, no normal flow, and Flather), and have specified throughout the paper which ones are used in the simulations (e.g. page 38 line 6, page 40 line 8).