ARCHER eCSE Final Report

eCSE Number:	eCSE02-3
eCSE Title:	*Scalable automated parallel PDE-constrained optimisation for dolfin-adjoint*
Date of Submission:	11 September 2015
PI and Co-Is:	Patrick E. Farrell Michele Weiland
Technical staff member(s):	Dominic Sloan-Murphy
Author(s) of this document:	Dominic Sloan-Murphy Patrick E. Farrell Michele Weiland
Project start date:	01 September 2014
Project completion date:	01 July 2015
Total number of funded project months:	8

Publishable Summary

1.1 Achievement of objectives

The project objectives were divided into four work packages, each with an associated deliverable. Success is measured by the quality and completeness of these deliverables.

Work package 1
Parallel nonlinear optimisation using PETSc
Deliverable
dolfin-adjoint with support for the optimisation algorithms in PETSc
Achievement
Successfully delivered:

- dolfin-adjoint with a user interface to PETSc/TAO optimisation algorithms employing petsc4py.
- dolfin-adjoint with refactored interface and test suite for optimisation packages IPOPT and Optizelle, consistent with the new PETSc functionality.

During the project, we realised that we had a major opportunity to dramatically improve the performance of the optimisation algorithms in PETSc. The optimisation algorithms, like all other open-source algorithms, formulate the optimisation problem in R^n, using the Euclidean inner product to measure distances, angles, convergence, etc. Dr Farrell realised that by modifying the optimisation algorithms to use an appropriate function-space inner product (e.g. L^2, or H^1), *mesh independence of the optimisation algorithm* could be achieved: the number of optimisation iterations would remain constant as the mesh is refined, rather than increasing with mesh size.

This was deemed of great interest and importance, and effort was therefore redirected from other WPs to support the implementation. As will be seen later, this work has had a dramatic impact on our ability to solve large problems. As a result of this redirection, the following additional deliverables were attained:

- Mesh-independent convergence support added to LMVM, BLMVM, NLS and NTR optimisation algorithms in PETSc.
- New Python bindings for associated functions.

Work package 2
Parallel I/O
Deliverable
dolfin-adjoint with parallel I/O checkpointing

Achievement
Successfully delivered:

- dolfin-adjoint with existing XML-based checkpointing replaced with a parallel HDF5 alternative.

Work package 3
Memory optimisation
Deliverable
Memory optimised dolfin-adjoint
Achievement
Not delivered: effort redirected to WP1.

Work package 4
Benchmarking and Report
Deliverable
Final eCSE report
Achievement
Collected in this document

1.2 Project description

The dolfin-adjoint package enables the automated optimisation of problems constrained by partial differential equations (PDEs). These problems are ubiquitous in engineering with examples including the design of wings to maximise lift, identification of the optimal placement of marine turbines for renewable tidal energy, the design of the cheapest bridge that will support its load, and the imaging of underground structures for petroleum exploration.

Prior to this project, dolfin-adjoint was limited to serial optimisation libraries with no concept of parallel linear algebra. Data was therefore required to be shared with all ranks and optimisation calculations performed redundantly. Now, the software is equipped with a Python-based interface to the parallel algorithms included in PETSc (the "Portable, Extensible Toolkit for Scientific Computation"), thereby eliminating the performance penalty associated with gathering the data and lifting the upper bound on the size of problems able to be considered.

The algorithms within the PETSc TAO (Toolkit for Advance Optimization) module were themselves extended to improve the efficiency of their solves. Specifically, the ability for a user to provide a Riesz map for the important LMVM, BLMVM, NTR, and NLS [1] solvers was added, allowing for use of a more appropriate inner product for a problem and enabling mesh-independent convergence, i.e. the number of iterations no longer depends on the complexity of the mesh.

This addition has had a dramatic effect on the performance of said solvers. Results for the Poisson mother problem [2] with a given mesh at various levels of random refinement, are provided in Table 1-1 below.

Number of random refinements	BLMVM Iterations required for original algorithm	BLMVM Iterations required for new algorithm
0	565	10
1	1095	4
2	Failed at 1463	4
3	-	4

Table 1-1: Effect of Riesz

As an illustrative example of the improvement in efficiency, Figure 1-1 and Figure 1-2 give a visualisation of the convergence for the Poisson mother problem after 40 BLMVM iterations on a given mesh without the Riesz map specified and 3 iterations with the map correctly set.

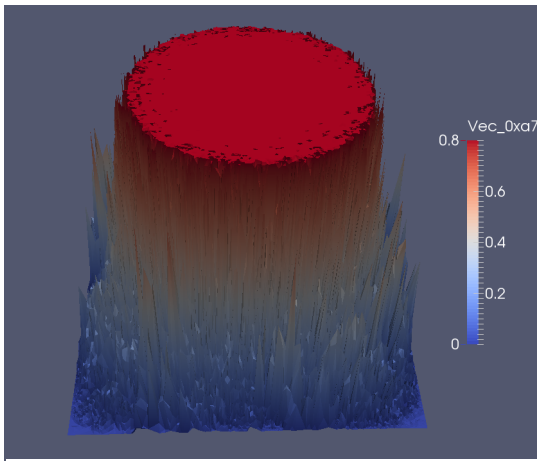

Figure 1-1: 40 iterations without Riesz

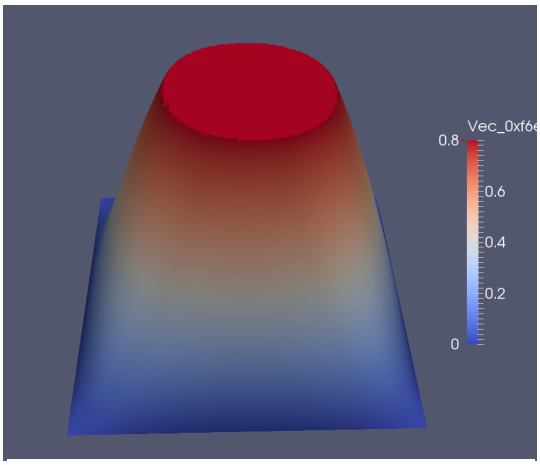

Figure 1-2: 3 iterations with Riesz

To the best of our knowledge, no other open-source package has managed to attain mesh independence in the bounded case (BLMVM), making this a significant achievement.

Additionally, this project investigated the checkpointing I/O performance of dolfin-adjoint, necessary during simulations with long time horizons on platforms with limited memory.

The previous implementation gathered all data to a single root process which would write out in XML format. This was replaced with a scalable HDF5-based solution built on work from a previous dCSE project involving the implementation of parallel I/O for the FEniCS software package [3].

Looking at a sensitivity analysis of the heat equation on a Gray's Klein bottle, the results in Figure 1-3 were observed on ARCHER.

```
# 480 MPI ranks, base mesh 256x256, 192 time steps
```

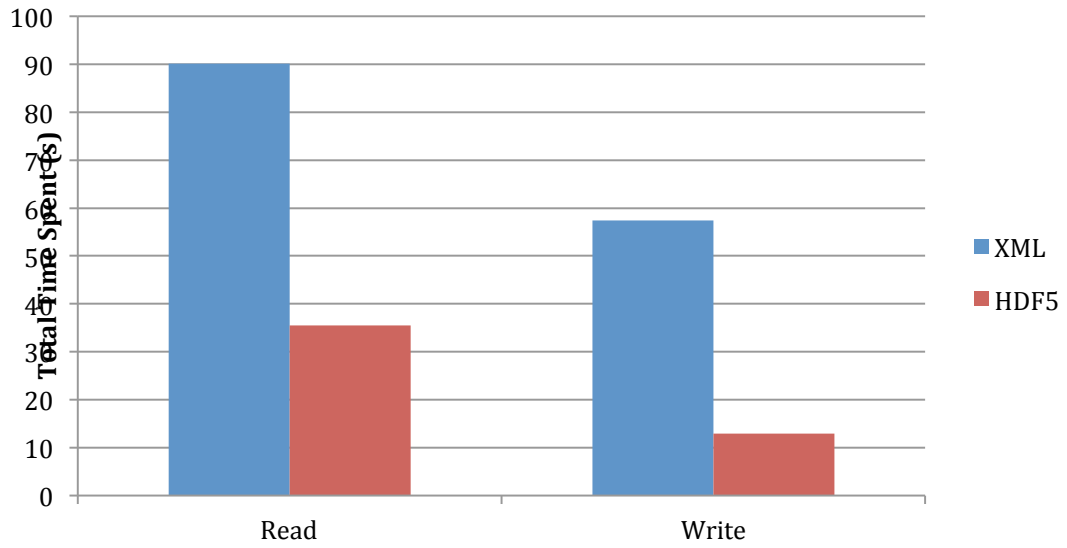

Figure 1-3: Klein Bottle Test – I/O Comparison

In this case, read times attain a speed up factor of approximately 3 with the new HDF5 approach while writes reach near 5 times faster. This opens the door to solving problems which have been bottlenecked by the limited I/O bandwidth of a single process or by the overheads of having to communicate all data to that single rank.

Links and References:

[1]: TAO 3.6 Users Manual, Retrieved 18 August 2015, from:
http://www.mcs.anl.gov/petsc/petsc-current/docs/tao_manual.pdf

[2]: Optimal control of the Poisson equation, Retrieved 18 August 2015, from:
http://www.dolfin-adjoint.org/en/latest/documentation/poisson-mother/poisson-mother.html

[3]: University Distributed CSE Project Report Expressive and scalable finite element simulation beyond 1000 cores, Retrieved 18 August 2015, from:
http://www.hector.ac.uk/cse/distributedcse/reports/UniDOLFIN/UniDOLFIN/dcse_richardson_wells.html

Project Website:
http://www.dolfin-adjoint.org/en/latest/

1.3 Summary of the software

All software outputs from the project have been merged into the master branches of the associated software, located at:

dolfin-adjoint (Primary)
https://bitbucket.org/dolfin-adjoint/dolfin-adjoint

PETSc
https://bitbucket.org/petsc/petsc

petsc4py Fork
https://bitbucket.org/petsc/petsc4py

A pull request was made to have changes merged into the main PETSc and petsc4py codebases. The work of the project has been accepted into the main "master" branch all codes involved.

The new functionality will be incorporated into a future PETSc release and made available to the entire user community.

On ARCHER, PETSc is fully supported and maintained by Cray. It is expected the version containing the output of this project will be installed as part of the regular update cycle of the Cray XE/XK Programming Environment. The enhanced petsc4py and dolfin-adjoint dependent on this release will subsequently be installed as central ARCHER packages.

Prior to this, ARCHER users will be able to access the software via the public repositories and employ the system development tools to install a local version for their own use.

Future science and impact

2.1 Performance Improvement

Name of simulation and code:	dolfin-adjoint: checkpointing I/O
Description of computational runs required:	
CPU-time before eCSE work per simulation (kAUs):	
CPU-time after eCSE work per simulation (kAUs):	
Estimate of overall financial saving/benefit:	
Comment (including any assumptions, number of users/runs estimates are based on, etc.):	The XML-based checkpointing was identified as a potential performance bottleneck following the scalability of the optimisation algorithms. An HDF5-based implementation was completed before this bottleneck had significant impact.

2.2 Additional Functionality

Name of simulation and code:	dolfin-adjoint PETSc/TAO
Description of computational runs required:	Poisson mother optimisation problem, 16x32 mesh, level 2 random refinement.
CPU-time before eCSE work per simulation (kAUs) to reach a given level of accuracy:	Failure at 1463 iterations
CPU-time after eCSE work per simulation (kAUs) to reach a given level of accuracy:	Convergence at 4 iterations

Estimate of overall financial saving/benefit:	
Comment (including any assumptions, number of users/runs estimates are based on, etc.):	New results can be achieved which could not have been achieved before as software is now using correct inner product via the Riesz mapping.

2.3 Sustainability of Software

Comment:

Action which has been made quicker (e.g. porting of code, integration of new feature):	Maintenance of test cases, troubleshooting/debugging.
Time taken before eCSE work per simulation:	
Time taken after eCSE work per simulation:	
Estimate of overall human effort saving/benefit:	
Comment (including any assumptions, number of users estimates are based on, etc.):	The new interface to PETSc is consistent with the interfaces to other dolfin-adjoint solvers. Users can easily switch between them with minimal modification to their test cases, saving much development time. The link to PETSc increases the potential user base significantly, increasing the level of expertise surrounding the software. Additional avenues of support are now available in the form of the very active PETSc mailing lists.

2.4 Usability of software

Action which has been made quicker (e.g. compilation of code, integration of new feature):	Development of test cases.
Time taken before eCSE work per simulation:	Significant work required to manually interface the package with petsc4py.

Time taken after eCSE work per simulation:	This effort is no longer required.
Estimate of overall human effort saving/benefit:	
Comment (including any assumptions, number of users estimates are based on, etc.):	The new interface between dolfin-adjoint and PETSc/TAO greatly increases the interoperability and general usability of the parallel optimisation algorithms.

2.5 Intrinsic value of the software

The new function-space-aware implementations in PETSc provide functionality not offered by any other open source project, significantly increasing the worth of the software within the academic and general open source community.

2.6 Enabled Science and Impact

The initial proposal for this project identified six concrete examples of the impact this project would have on the world scientific community:

- *Eddy parameterisation in quasigeostrophic turbulence* – letter of support from Dr James Maddison (Edinburgh).
- *Identification of triggering and limit cycles in thermoacoustic systems* – letter of support from Dr Matthew Juniper (Cambridge).
- *Inverse problems in cardiac electrophysiology* – letter of support from Dr Molly Maleckar (Simula).
- *Automated urinalysis for diagnosis of urinary tract infections (UTIs)* – letter of support from Dr Irwin Zaid (Speckle Technologies).
- *Layout optimisation of tidal turbines for renewable energy* – letter of support from Dr Matthew Piggott (Imperial College London).
- *Shape optimisation of acoustic waveguides* – letter of support from Dr Stephan Schmidt (Würzburg).

The most common limiting factor identified in these cases was the all-gathering of the parameter vector onto each core. This has been eliminated by the new interface to the parallel algorithms in PETSc. The software now allows for far larger parameter vectors, enabling new science in these communities.