4 Technical Report (publishable)

Abstract

TPLS is a freely available CFD code which solves the Navier-Stokes equations for an incompressible two-phase flow. The level set method enables sharp changes in interfacial topology and is therefore ideally suited to gaining an understanding of problems of interest to the oil and gas industry as well as looking at the implications of the inherent instability of the interface in a two-phase system. This eCSE project aimed to improve the architecture and performance of the software, reduce overall runtime, and improve usability and maintainability, opening up the software to a large user community. The main aims of this project were to introduce TPLS 3.0 with: i) Implementation of 3D domain decomposition, ii) Replacing all SOR/Jacobi hand-coded routines with PETSc and iii) Implementation of density-contrast flows and implementation of a choice of boundary conditions. This eCSE work has resulted in a significant re-factoring of the TPLS software, improving maintainability by increasing the use of error checking, debugging options and the use of modules, functions and subroutines to minimise code duplication. This should enable future development work to be significantly quicker and easier to debug, as well as providing more runtime options, moving away from extensive use of hard-coded compilation options. As a result we are now able to provide TPLS 3.0 as a package available on ARCHER for all users TPLS 3.0 is released as opensource, available from Sourceforge.

1. Introduction

TPLS is a powerful and efficient 3D Direct Numerical Simulation (DNS) flow solver to simulate multiphase flows at unprecedented detail, speed and accuracy.

This flow solver has been developed by Lennon Ó Náraigh (School of Mathematical Sciences, University College Dublin), Prashant Valluri (School of Engineering, University of Edinburgh), David Scott, Toni Collis and Iain Bethune (the Edinburgh Parallel Computing Centre at the University of Edinburgh) and Peter Spelt (Université de Lyon1, Claude Bernard) under the aegis of several HECToR computer time grants and funding from EPSRC, the HECToR dCSE programme administered by NAG and the ARCHER eCSE programme administered by EPCC. The solver has clocked over 1300 downloads around the world. We summarise the versions so far in Table 1.

Table 1: Versions of TPLS, their features, parallel performance and issues

Version	Features	Performance and constraints
TPLS 0.0	• Primitive version of code, using hand-coded SOR and Jacobi routines for the linear solves • Serial I/O	• Poor scalability • Access to density contrast flows (branch version) • Choice of boundary conditions (inlet/outlet or periodic)
TPLS 1.0	• Hand-coded routines substituted by PETSc routines in pressure-solver • Serial I/O	• scalability improvement (speed-up = 10% on 1024 cores) • density matched flows only (liquid/liquid)

		• inlet/outlet boundary conditions only
TPLS 2.0	• PETSc routines in pressure-solver • Parallel I/O	• Scalability improvement (speed-up = 10.9x on 3072 cores) • Density matched case only • Inlet/outlet boundary conditions only

This eCSE work concerns crucial improvements of TPLS 2.0 necessary to enable simulation of larger-scale flows relevant to industry involving large density contrasts (as in gas/liquid systems) and larger simulated domains in order to accelerate its industrial uptake.

The main aim of this work has been to introduce TPLS 3.0, which (among other things) will bring together features from branch versions of the code, and remove the remaining drawbacks. In particular: i) Implementation of 3D domain decomposition, ii) Replacing all SOR/Jacobi hand-coded routines with PETSc and iii) Implementation of density-contrast flows and implementation of a choice of boundary conditions (inlet/outlet or periodic).

2. Objectives and Success Metrics

***Objective 1**: Full 3D domain decomposition, leading to improved strong scaling (80% parallel efficiency on 64 ARCHER nodes for a 5.6 million element problem)*

This is important for real gas/liquid turbulent systems that demand a higher resolution and larger number of cores – 3D decomposition would allow us to simulate non-linear late time flow behaviour within a single ARCHER job (needs multiple restarts with TPLS 2.0)

***Outcome:** A full 3D domain decomposition has been implemented successfully. Surprisingly, the parallel efficiency on 64 nodes is a disappointing 20% but the speed-up relative to the 2D code is greater than 2.*

***Objective 2:** Replace all SOR solvers with PETSc[2] (50% speedup in each compute routine on 64 nodes)*

This is necessary to improve the scalability of TPLS 2.0.

***Outcome:** All the three momentum solvers, which were SOR-based, have been implemented with PETSc. However, the interface detection solvers have not. The measurements were actually carried out on 32 nodes using the density contrast code on an 11 million grid point domain. The PETSc (Krylov) momentum solvers ran at only 2/3 the speed of the SOR solvers. The PETSc pressure solver ran at the same speed as the corresponding SOR solver. The expected speed-ups were based on the speed-up produced by the introduction of a PETSc solver into the*

[2] PETSc is the Portable, Extensible Toolkit for Scientific Computation, see https://www.mcs.anl.gov/petsc

pressure calculation during a previous project. Unfortunately, the implementation of the previous solver was faulty and correcting the error led to a substantial decrease in speed.

Objective 3: *Implementation of density-contrast flows (demonstration gas/liquid flow calculation using 3D PETSc code)*

- *TPLS 3.0 with density contrasts should give the same performance as TPLS 2.0 without density contrasts and at least 7 times as fast as the TPLS 0.0 branch for a density ratio of 1000.*
- *Validate against the reference case of Trygvasson (1988) for Rayleigh-Taylor instability – a canonical density-contrast two-phase instability.*

This is an essential improvement to encourage industry uptake.

Outcome: *Density contrast flows have been successfully implemented. A gas/liquid flow calculation has also been demonstrated using this TPLS 3.0 solver. A comparison with TPLS 2.0 has not been carried out because the result would be uninformative. It was noticed that TPLS 2.0 had incorporated a faulty pressure solver which has a significant impact on the execution speed of the code. This has been corrected in TPLS 3.0. For the 11 million point problem running on 32 nodes TPLS 3.0 was 18% faster than TPLS 1.0 when using the SOR solvers. The PETSc solvers were 39% slower. The implementation has been validated for the case of the Rayleigh-Taylor instability.*

Objective 4: *Release TPLS 3.0*

This is necessary to ensure dissemination of the solver to the wider audience.

Outcome: *TPLS 3.0 has been released via Sourceforge on 08 December 2017 under the aegis of the UK-wide Special Interest Group Meeting on Multiphase Flows and Transport Phenomena. This was attended by around 55 experts and students from both UK industry and academia. The total number of downloads so far have been 1361 (55 downloads more since TPLS 3.0 release on Sourceforge)*

The project was a success with most of the objectives met. Key highlights include:

- A fully implemented 3D decomposition and momentum solvers strengthened by implementing PETSc routines (and removing older SOR routines). *This resulted in TPLS 3.0 solver being at least two times faster than the TPLS 2.0.*
- Fully capable of efficiently handling density contrast flows, see Figure 1. Density ratios of 1000 (similar to water/air) are now easily achievable at top-speeds. Most available solvers fail for high density ratios due to the excessive computational requirement and are

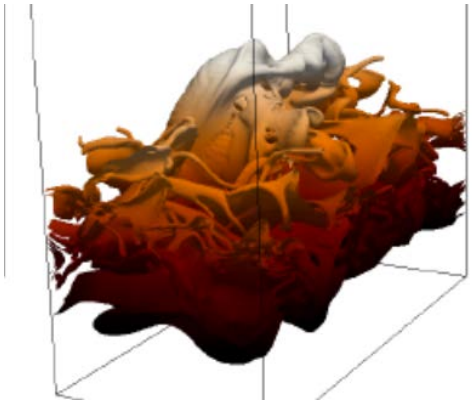

Figure 1: Complex "turbulent" interface of a two-phase flow simulated at moderate density ratios and flowrates.

therefore unable to simulate industrial scale flows. *The fact that TPLS now offers high-density ratio solvability alongside its proven ultra-high resolution industrial-scale parallelism is a major improvement. This will therefore widen the application of TPLS to industry.*

3. TPLS 3.0 Performance

The project set out to improve the performance of the code

- by changing from a 2D to a 3D decomposition of the domain and
- by replacing the hand crafted iterative solvers (mostly Jacobi plus SOR solvers) present in the code by solvers from the PETSc library (i.e. by preconditioned Krylov solvers).

It also set out to extend the range of problems that could be studied using the code by allowing the different fluid components to have differing densities. This was achieved by integrating existing, bespoke, 2D code into the more general framework of the publicly released TPLS code (converting it to 3D in the process).

Finally, the project also set out to provide the user with a choice of inlet or periodic boundary conditions in the direction of flow.

In the past, much work related to TPLS has resulted in more or less stand alone code. An unstated objective was to encourage those people working on developments related to TPLS to do so in the context of the public TPLS code by working on branches which could then be more easily merged into the public code. The code has been written so that it is possible to select at run time (solver by solver) whether a 3D reimplementation of the original solver or a Krylov solver is used. It is hoped that this will lower the barrier to the code being used as a basis for future developments.

The objectives listed above proved to be ambitious although most of the work was carried out. The main reason for this was that it took much longer than expected to get the PETSc solvers working properly. Exact work undertaken is described below.

The TPLS code contains six iterative solvers:

- three momentum solvers (one for each component)
- a pressure solver
- a levelset solver
- a diffuse interface method solver

Four of these six have now been implemented using PETSc solvers. The interface solvers have not been reimplemented. They contain peculiarities that would have complicated their reimplementation. Furthermore, testing showed that the interface solver took only a small part of the execution time compared to the momentum solvers. As a result, the team decided that the available effort would be better spent on other goals of the project.

As a team, we also decided that for research purposes the availability of a periodic boundary condition was more important than the availability of an inlet boundary condition. Hence, priority was given to the implementation of periodic boundary conditions. This goal was achieved. However, a version of the code implementing an inlet boundary condition also exists as a branch. This branch is close to being ready to release as a separate download but is not currently publicly available.

Full 3D domain decomposition was implemented and the density contrast code has been merged into the 3D code, so most of the planned implementation work has been achieved. The 3D domain decomposition was achieved by exploiting the capabilities of the PETSc toolkit which made this much easier to do than would otherwise have been the case. The parallel efficiency on 64 nodes for a 5.6 million (256x144x152) element problem is a disappointing 20% (see the graph in Figure 2) but the speed-up relative to the 2D code is greater than 2.

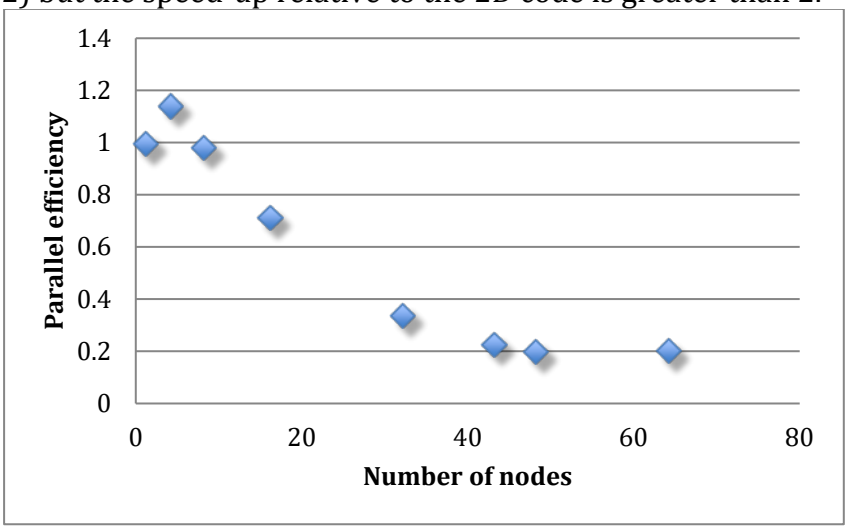

Figure 2: Scaling as a result of full 3D decomposition.

While a 20% speed up is obviously lower than what we envisaged, during the project we realised that implementation of the pressure solver in a previous project was faulty and led to an unrealistic expectation regarding the speed-up that the current reimplementation would achieve. Correcting the fault approximately doubled the execution time of the solver.

Doubling the size of the domain improves the scaling behaviour as shown by the Figure 3. For this larger problem the parallel efficiency is 0.65 on 64 nodes.

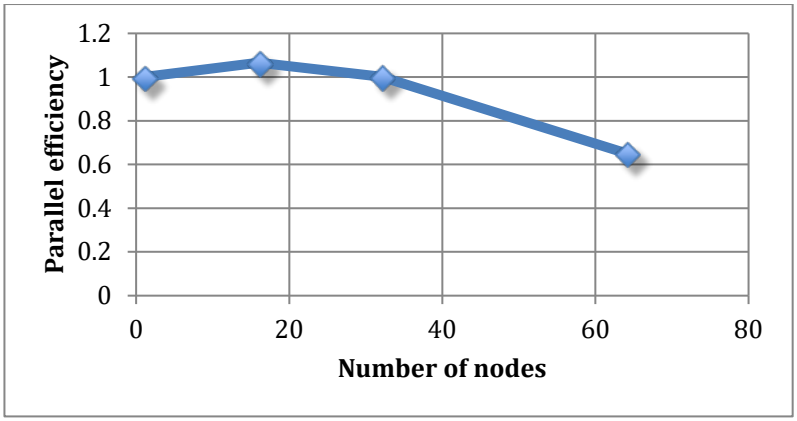

Figure 3: Scaling of TPLS 3 for 11 Million Point Domain

PETSc allows one to select which of the available Krylov solvers is used at run time and one can use this flexibility to tune performance. Unfortunately, the hope that a clever choice of solver will radically improve the performance of the code is unfounded as the matrix orientated approach adopted means that a new matrix has to be constructed for each solver at each time step and this construction process takes about as long as obtaining the approximate solution once the matrix is available. The performance figures quoted below were obtained using a GMRES solver and ILU preconditioner for the momentum equations and a MINRES solver with a SOR preconditioner (with omega = 1.5) for the momentum equations.

The three momentum solvers have been implemented with PETSc, the interface detection solvers have not. Measurements to determine the speed-up achieved were carried out on 32 nodes using the density contrast code on an 11 million point grid. The PETSc (Krylov) momentum solvers ran at only 2/3 the speed of the SOR solvers. The PETSc pressure solver ran at the same speed as the corresponding SOR solver. These speed-ups fall short of the expected speed-ups but these were based on the speed-up produced by the introduction of a PETSc solver into the pressure calculation during a previous project. Unfortunately, as explained above, the previous implementation of the pressure solver was faulty and correcting the error led to a substantial decrease in speed.

The speed-ups reported above were not simple to determine as the SOR solvers terminate after a fixed number of iterations whilst the termination conditions of the PETSc solvers depend on the behaviour of the preconditioned residual norm. When setting up the experiments code to compute true residual norms was activated and the convergence criteria of the PETSc solvers were adjusted to try to match the true residual norms of the PETSc solvers to those of the SOR solvers.

Density contrast flows have been implemented and the implementation has been validated for the case of the Rayleigh-Taylor instability. Some experimental results have been obtained for an 11 million (512x144x152) grid point domain. Running on 32 nodes TPLS 3.0 was 18% faster than TPLS 0.0 when using the SOR solvers. The PETSc solvers were 39% slower. Also some scaling results have been obtained. The density ratio was 30 and gravity was zero.

Number of nodes	**1**	**16**	**32**
Parallel efficiency (SOR)	1	1.1	0.5
Parallel efficiency (Krylov)	1	2.0	1.0

4. Conclusions

As a result of this eCSE project, TPLS 3.0 solver outperforms its predecessor in terms of both speed-up and resolution. It remains one of the few ultra-high resolution DNS solvers for two-phase flows available globally. The new 3D decomposition and PETSc routine implementations make it highly parallelisable and at the same time customisable for bespoke supercomputing architectures. This, therefore, makes it popular (evidenced by the high number of downloads) amongst academics and industry. TPLS 3.0 has been successfully tested already in industrial, Irish, American and UK supercomputing systems.

TPLS 3.0 is now available free to use under the BSD Licence and can be obtained from: http://sourceforge.net/projects/tpls/

Key features are:

1. Highly parallelisable using an MPI architecture;
2. Optimised on ARCHER (> 1536 Cores);
3. Fortran coding with PETSc subroutines and parallel-NETCDF I/O.
4. Density contrast flows (liquid/gas flows, density ratios > 1000/1)
5. 3D Domain decomposition and PETSc momentum solvers for efficient parallelism

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