Science & Technology Facilities Council

Technical Report

RAL-TR-2018-001

Several parallel-in-time methods to solve partial differential equations

P Gambron

February 2018

Enquiries concerning this report should be addressed to:

RAL Library
STFC Rutherford Appleton Laboratory
Harwell Oxford
Didcot
OX11 0QX

Tel: +44(0)1235 445384
Fax: +44(0)1235 446403
email: libraryral@stfc.ac.uk

Science and Technology Facilities Council reports are available online at: http://epubs.stfc.ac.uk

ISSN 1358-6254

Several Parallel-in-time Methods to Solve Partial Differential Equations

Philippe Gambron *

Science and Technology Facilities Council, Rutherford Appleton Laboratory, Scientific Computing Department, Harwell Campus, Harwell Oxford, OX11 0QZ, United Kingdom

January 29, 2018

Abstract

We introduce a parallel-in-time method solving partial differential equations by alternating between a coarse and a fine grid. We present as well the classical algorithm Parareal and a more recent development, Paraexp.

1 Introduction

Increasing the spatial resolution of a large-scale simulation also requires a smaller time step. This can be interpreted as the necessity to prevent the Courant number from becoming prohibitively large. As a consequence, a large high resolution simulation using a time-marching approach will face a bottleneck in the time direction. A remedy can consist in introducing a degree of parallelism in the time direction as well. This can also be useful when the parallelism in spatial directions is saturated, that is when the overhead is such that more parallelism no longer brings any performance improvement. As we will see, parallelism-in-time is not optimal but it is the only way to obtain a further performance improvement in the situations we have described.

*philippe.gambron@stfc.ac.uk

2 Using a coarse grid in space-time

We propose a parallel-in-time method to solve partial differential equations by alternating, in a way inspired by multi-grid methods, between a resolution on a coarse grid in space-time and a resolution on a fine one where several time steps are processed simultaneously and boundary conditions are obtained from the coarse grid. Each subgrid is handled by a different process and the information exchange between the halos is replaced by updating the average values located at the vertices and interpolating between them. This leads to an improvement in the performance and the parallel scalability.

The unstructured mesh is split in different parts which will be handled by different processes. The equation is solved, with a limited precision, on each part of the mesh by the Gauss-Seidel method where the identity has been added to the equation in order to preserve the diagonal dominance and the sparse matrix is modified at every iteration to handle the non-linearity. During this step, the boundary values for several time points are kept constant and are provided by an interpolation from a coarse grid in space and time which will be updated once this step is complete (fig. 1). Therefore the usual copy of the halos is replaced by an interaction with a coarse grid after several time steps have been processed. This operation is repeated a few times, till a moderate precision has been reached. Finally, the system is solved the usual way, to a high precision, by copying the halos at every iteration. However, a limited number of the initial iterations, alternating between a coarse resolution and a fine one where boundary conditions are kept constant and where all the information exchange between the nodes occurs by interacting with the coarse grid, is able to significantly decrease the number of iterations, done the classical way, necessary to reach the final solution. We do no address the issue of causality since we use the Gauss-Seidel method, which places time and space on an equal footing. This is especially useful for PDEs mixing temporal and spatial derivatives like the shallow water equations.

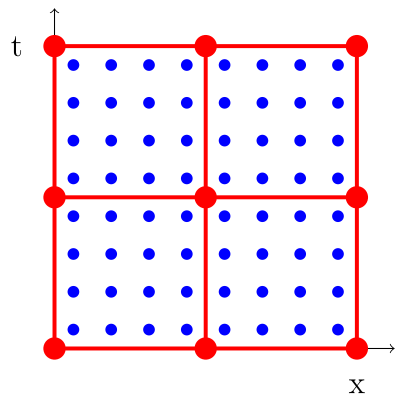

Figure 1: The boundary conditions for several time steps are interpolated from the values on a coarse grid in space and time.

The iterations alternating between a coarse grid and a fine one improve the performances as can be seen with a lid-driven cavity flow with obstacles on a mesh consisting of $\sim 4\ 10^6$ space-time cells (fig. 2). This example is parallel in space and time. The size of the spatial subdomains are the relevant fraction of the physical domain whereas the number of time steps in a time slice is much more modest, as we are about to see. The same case can be used to demonstrate that the performance improvement is maximal for moderate sizes of the time slices (fig. 3) and that the scaling is better than with OpenFoam (fig. 4).

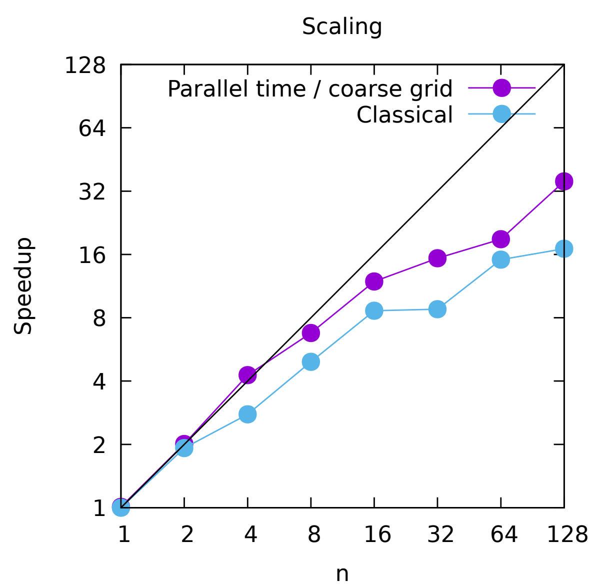

Figure 2: Effect of the interactions with the coarse grid

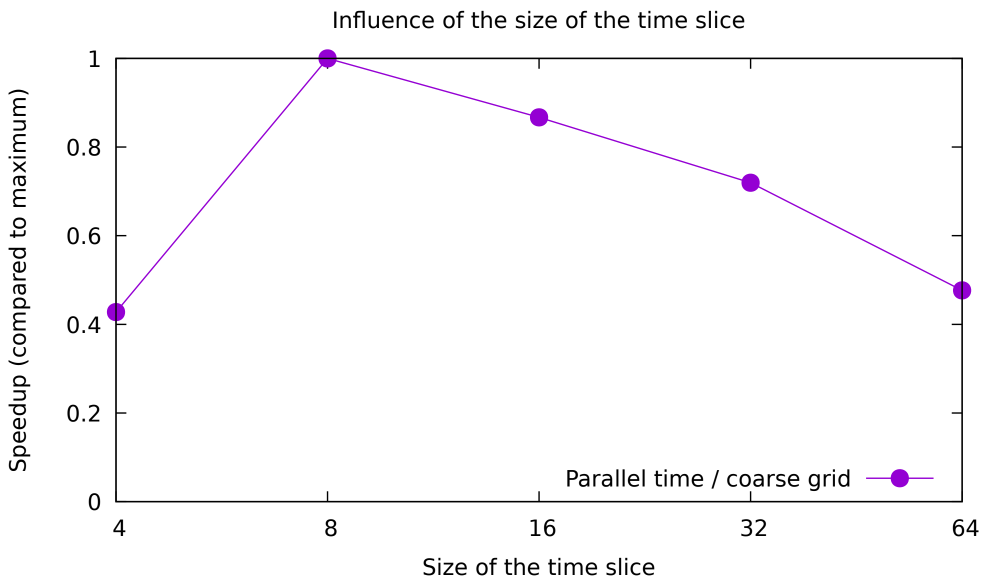

Figure 3: The speedup is optimal for time slices of a moderate size.

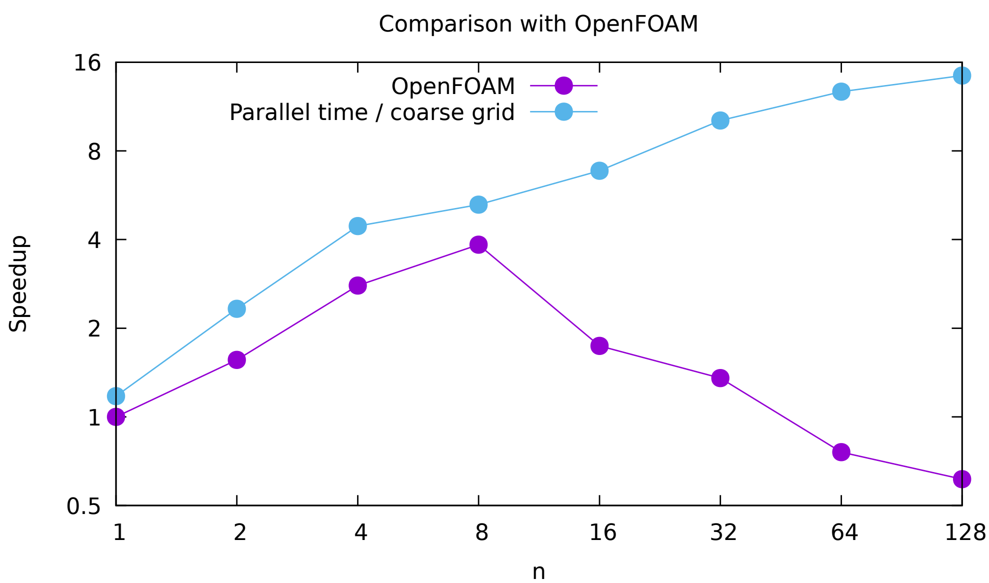

Figure 4: Comparison with OpenFOAM

The comparison with OpenFOAM is more delicate because of the multigrid aspect of its solver. As a consequence, for very large meshes, OpenFOAM becomes dramatically faster. However, the method used seems unstable because it requires a time step an order of magnitude smaller in order to converge. That property makes our software much faster on meshes that are not too large, that is not larger than the one we are considering here.

3 Parareal and MGRIT

Parareal solves an initial value problem on different time intervals in parallel by using initial values coming from the resolution of a coarse problem and then bringing a correction to these initial values till a smooth solution is found on the entire time domain [1]. As a consequence, causality will be taken into account. A comparison of the scaling of parareal and the one obtained with a spatially parallel resolution illustrates the true cost of parallelism in time (fig. 5). This has been obtained in solving the heat equation with $\sim 10^6$ space-time cells.

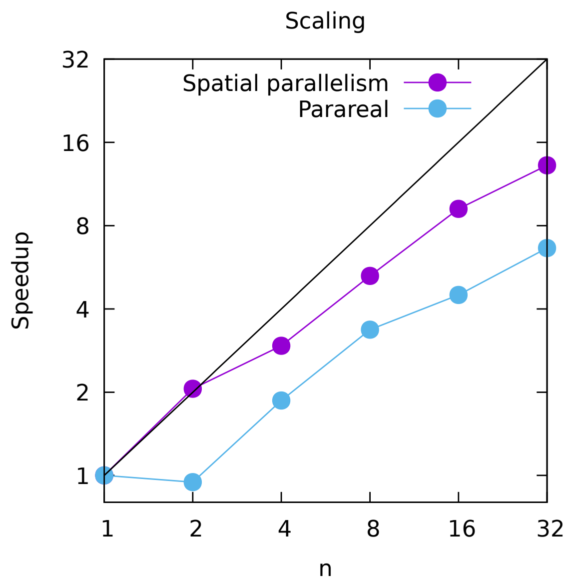

Figure 5: Comparison between the scaling obtained with Parareal and the one obtained with a spatially parallel method

Parareal can be seen as a two-level multi-grid method. MGRIT (Multi-Grid Reduction In Time) is a generalisation of the algorithm making it fully multi-grid, on many coarseness levels [2].

4 Paraexp

Paraexp solves a PDE on several time intervals in parallel by solving an homogeneous equation with a non-zero initial value and an inhomogeneous one with a vanishing initial value and combining their solutions. The determination of the initial values at the beginning of each time interval is done in parallel and just once. Subsequently, one makes use of those initial values but only once and again in parallel with the help of exponential integrators. As a consequence, unlike with Parareal, the solution does not have to be computed again once the initial values have been amended. As can be seen in fig. 6, the scaling is excellent. This has been obtained solving the heat equation in the same conditions as previously.

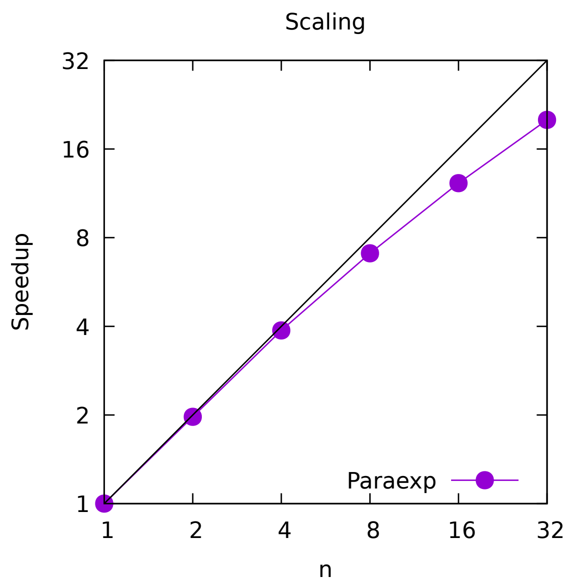

Figure 6: Scaling obtained with Paraexp

Unfortunately, this algorithm is limited to linear problems. This is necessary to combine the different pieces used to construct the solution. Nevertheless, recent attempts have been made to solve non-linear PDEs with this method by placing the non-linearity in the source term [4].

5 Future directions

A method that seems promising would be a combination of the MGRIT algorithm with spatial parallelism implemented in a fully multi-grid way, on several levels. In that case, the spatial multi-grid aspect should be algebraic in order to accommodate an unstructured grid. To our knowledge, this has never been done.

6 Conclusion

Parallel-in-time methods are non-optimal but necessary. They are the only remaining way to improve performances in the case of, for example, large scale high-resolution simulations. We have introduced a method where space and time are treated in the same way since it relies on the Gauss-Seidel algorithm and introduced two other methods, which take causality into account, Parareal, the classical one, as well as a new development, Paraexp, making use of exponential integrators and computing the effect of the initial value, at the beginning of each time step, just once, in a direct way.

Acknowledgments

This work has been funded by EPSRC under grant EP/K038303/1. I would like to express my gratitude to Prof Beth Wingate, from Exeter University, for her suggestions.

References

[1] J. Lions, Y. Maday and G. Turinici; *A 'Parareal'-in-time Discretization of PDE's*; Comptes Rendus de l'Acadmie des Sciences Series I Mathematics, Vol. 332, Num. 7, p 661-668

[2] S. Friedhoff, R. D. Falgout, T. V. Kolev, S. P. MacLachlan and J. B. Schroder, *A Multigrid-in-time Algorithm for Solving Evolution Equations in Parallel*; presented at the Sixteenth Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, United States, 17-22 Mar, 2013

[3] M. J. Gander and S. Guttel; *Paraexp: a Parareal Integrator of Linear Initial Value Problems*; preprint available at `https://www.unige.ch/ gander/Preprints/paraexp.pdf`

[4] G. L. Kooij, M. A. Botchev and B. J. Geurts; *A Block Krylov Subspace Implementation of the Time-parallel Paraexp Method and its Extension for Nonlinear Partial Differential Equations*, `arXiv:1509.04567v1`