Direct FEM parallel-in-time computation of turbulent flow

Johan Jansson, Johan Hoffman

November 1, 2017

Abstract

...

1 Introduction

Large-scale supercomputing with distributed memory has become pervasive in many scientific disciplines, with Computational Fluid Dynamics (CFD) as one of the main application areas. This has enabled scientists to investigate problems several orders of magnitude larger than would be possible on a standard workstation, which was the standard computational resource for most scientists for the past decades.

1.1 Parallel in time methods

The dominant paradigm to solve a large scale space-time partial differential equation (PDE) is partition the spatial domain, and using distributed linear algebra algorithms to solve the global problem in parallel. This paradigm has shown good parallel scaling, up to hundreds of thousands of computing cores, or processing elements (PE) but there is an inherent bottleneck in that the time dimension is not parallelized. This is a severe bottleneck since the spatial cell size and time step size need to be typically proportional for stability reasons. There is also a limit to how few spatial degrees of freedom can be assigned to each PE, and this imposes a limit for the wall-clock time required to solve one time step.

As a possible remedy to this problem, Lions et. al. proposed the parareal method [1] which is a multilevel approach for parallelizing the time dimension, i.e. a parallel-in-time method. The method in parallel timesteps partitions of the time interval with the finest timestep size, and uses a coarse timestepping of the global problem to couple the local partitions. The coarse solver which is computationally cheap is executed in serial and provides the initial condition for the fine solver, which is run in parallel. The method then iterates through this procedure until it converges to the solution of the initial PDE, under certain

conditions. Good performance of the parareal algorithm has been demonstrated for a wide range of problems such as molecular dynamics [2] and stochastic ordinary differential equations [3] with significant improvements in timing (see also the references provided by Minion [4]).

Over recent years, the parareal algorithm has been applied to the Navier-Stokes equations and its performance has been analyzed and studied in many works. Fischer et. al. [5] applied parareal to the laminar regime of the Navier-Stokes equations. In the manuscript by Croce et. al., [6] a combined parareal and domain-decomposition approach is used to successfully solve the three-dimensional, time-dependent, incompressible Navier-Stokes equations. The simulations, which were run on up to 2048 cores, show the applicability of a parallel-in-time method to challenging cavity flow problems. For the quasi-2D flow problem, parareal converges rapidly. However, the observed convergence rate stagnates for the 3D case when the number of time-intervals taken by the parareal solver becomes larger. The stability issues observed by Croce et. al. coincide with the stability issues that parareal faces when applied to hyperbolic and convection-dominant problems, see for example [7, 8, 9]. Numerical results shown in Steiner et. al. [10] confirm that the convergence of parareal worsens as the Reynolds number increases and the flow becomes more and more convection-dominated.

1.2 Turbulent flow

Recently, there has been active research in the field of time-parallel methods applied to turbulent flow problems. Reynolds-Barredo et. al. [11] showed convergence of the parareal algorithm on large scale numerical simulations of turbulent plasmas. However, as discussed in Wang et. al. [12] and references therein, the computational cost of most parallel-in-time methods, including parareal, is quadratic with respect to the time domain length and therefore the performance of parareal deteriorates as the length of the time domain increases. This poor scalability is argued to be due to the sensitivity of the algorithm to chaos, i.e. a small perturbation in the early partitions of parareal can result in a large difference in the solution estimate and the converged solution. Wang et. al [12] reformulated the turbulent problem by relaxing the initial condition in order to make parallel-in-time method stable. The system with a relaxed initial condition does not suffer from the ill-conditioning present when simulating turbulent flows with a fixed initial value. This relaxation is justified under two assumptions, namely that (1) the quantities of interest are stable flow statistics when the system has reached quasi-steady state and (2) the system is ergodic, i.e. that the initial condition has small, or no influence, on the solution after a transient period has elapsed. Both these assumptions allow for efficient and scalable parallel-in-time algorithms.

We have over a number of years developed methods, algorithms and software for efficiently predicting gross quantities of turbulent flow, such as aerodynamic drag and lift coefficients [13, **?**]. We here aim to apply some of the concepts in our computational methodology to investigate the potential for exploiting

parallel-in-time scalability for turbulent flow. In this paper, we propose an embarrassingly parallel implementation of the Direct FEM-Simulation (DFS) [13] method for when the system satisfies the same two properties discussed in Wang et. al. [12]: That the quantity of interest is a stable flow statistics after the system has reached steady state and that the system satisfies the ergodicity property. The DFS method is a stabilized finite element method with automatic turbulence modeling based on the residual of the equations. We demonstrate the parallel-in-time DFS method on a model problem for the incompressible Euler equations approximating flow past a cube in a channel with high Reynolds number, where the quantity of interest is the drag coefficient of the cube. We show that after an initial transient time the quantity of interest (here the drag coefficient) satisfies the ergodicity property, and we conclude by showing the speedup of using the DFS method versus a purely space parallel method with the same accuracy.

2 Computation of turbulent flow

The incompressible NS equations for a unit density Newtonian fluid with constant kinematic viscosity $\nu > 0$ enclosed in a volume Ω in $\mathbb{R}^3$ (where we assume that Ω is a polygonal domain), together with suitable boundary conditions, take the form:

$$R(\hat{u}) = 0, \quad \text{in } Q = \Omega \times I, \tag{1}$$

for $\hat{u} = (u, p)$, with $u(x,\, t)$ the *velocity vector* and $p(x,\, t)$ the *pressure* at $(x,\, t)$, with $I = (0, T)$ a time interval, and $R(\hat{u}) \equiv \bar{R}(\hat{u}) - (f, 0) = (\bar{R}_1(\hat{u}), \bar{R}_2(u)) - (f, 0)$ the *residual*, where

$$\begin{aligned} \bar{R}_1(\hat{u}) &= \dot{u} + u \cdot \nabla u + \nabla p - \nu \Delta u \\ \bar{R}_2(u) &= \nabla \cdot u. \end{aligned} \tag{2}$$

The main existence result available is due to Jean Leray, who in 1934 proved the existence of a so-called *weak solution* (or *turbulent solution* in the terminology of Leray) [14], which is a solution satisfying (1) in an average sense; $\hat{u}$ is a weak solution if

$$((R(\hat{u}), \hat{v})) = 0, \tag{3}$$

for all *test functions* $\hat{v}$ in a *test space* $\hat{V}$ with norm $\|\cdot\|_{\hat{V}}$, consisting of suitable differentiable functions, $R(\hat{u})$ is assumed to belong to a space dual to $\hat{V}$, and $((\cdot, \cdot))$ denotes a duality pairing.

In a turbulent flow we may expect an extreme pointwise perturbation growth, and thus pointwise uniqueness is too much to ask for. On the other hand, some aspects of turbulent flow are more stable. Typically we expect various types of mean value output in a turbulent flow to be more stable than pointwise output.

2.1 Output Sensitivity and the Dual Problem

Suppose that the *quantity of interest*, or *output*, related to a given weak solution $\hat{u}$ is a scalar quantity of the form

$$M(\hat{u}) = ((\hat{u}, \hat{\psi})), \tag{4}$$

which represents a mean value in space-time, where $\hat{\psi} \in L_2(Q)$ is a given weight function. The output could e.g. be the drag or lift coefficient in a bluff body problem, for which the weight $\hat{\psi}$ is a piecewise constant function in space-time. More generally, $\hat{\psi}$ may be a piecewise smooth function corresponding to a mean value output.

We now seek to estimate the difference in output between a weak solution $\hat{u} = (u, p)$ and a finite element approximation $\hat{U} = (U, P)$. To this end, we introduce the following linearized dual problem of finding $\hat{\varphi} = (\varphi, \theta) \in \hat{V}$, such that

$$a(\hat{u}, \hat{U}; \hat{v}, \hat{\varphi}) = ((\hat{v}, \hat{\psi})), \quad \forall \hat{v} \in \hat{V}_0, \tag{5}$$

where $\hat{V}_0 = \{\hat{v} \in \hat{V} : v(\cdot, 0) = 0\}$, and

$$\begin{aligned} a(\hat{u}, \hat{U}; \hat{v}, \hat{\varphi}) \equiv ((\dot{v}, \varphi)) + ((u \cdot \nabla v, \varphi)) + ((v \cdot \nabla U, \varphi)) \\ +((\nabla \cdot \varphi, q)) - ((\nabla \cdot v, \theta)) + ((\nu \nabla v, \nabla \varphi)) \end{aligned}$$

with u and U acting as coefficients, and $\hat{\psi}$ is given data.

This is a linear convection-diffusion-reaction problem in variational form, u acting as the convection coefficient and ∇U as the reaction coefficient, and the time variable runs 'backward' in time with initial value $(\varphi(\cdot, T) = 0)$ given at final time T imposed by the variational formulation. The reaction coefficient ∇U may be large and highly fluctuating, and the convection velocity u may also be fluctuating.

Choosing now $\hat{v} = \hat{u} - \hat{U}$ in (5), we obtain

$$((\hat{u}, \hat{\psi})) - ((\hat{U}, \hat{\psi})) = a(\hat{u}, \hat{U}; \hat{u} - \hat{U}, \hat{\varphi}) = -((R(\hat{U}), \hat{\varphi})).$$

We may estimate the difference in output as follows:

$$|M(\hat{u}) - M(\hat{U})| \leq C \|hR(\hat{U})\|_{L(\Omega)} \|\hat{\varphi}\|_{H^1(Q)}, \tag{6}$$

using standard interpolation error estimates [**?**].

By defining the *stability factor* $S(\hat{u}, \hat{U}; \hat{\psi}) = \|\hat{\varphi}\|_{H^1(Q)}$, we can write

$$|M(\hat{u}) - M(\hat{U})| \leq C \|hR(\hat{U})\|_{L(\Omega)} S(\hat{u}, \hat{U}; \hat{\psi}). \tag{7}$$

Depending on $\hat{\psi}$, the stability factor $S(\hat{u}, \hat{U}; \hat{\psi})$ may be small, medium, or large, reflecting different levels of output sensitivity, where we expect $S(\hat{u}, \hat{U}; \hat{\psi})$ to increase as the mean value becomes more local. Estimating $S(\hat{u}, \hat{U}; \hat{\psi})$ using a standard Grönwall type estimate of the solution $\hat{\varphi}$ in terms of the data $\hat{\psi}$ would give a bound of the form $S(\hat{u}, \hat{U}; \hat{\psi}) \leq Ce^{KT}$, where C is a constant and

K is a pointwise bound of $|\nabla U|$. In a turbulent flow with $Re = 10^6$ we may have $K \sim 10^3$, and with $T = 10$ such a Grönwall upper bound of $S(\hat{u}, \hat{U}; \hat{\psi})$ would be of the form $S(\hat{u}, \hat{U}; \hat{\psi}) \leq Ce^{KT} \sim e^{10\,000}$, which is an incredibly large number.

However, computing the dual solution corresponding to drag and lift coefficients in turbulent flow, we find values of $S(\hat{u}, \hat{U}; \hat{\psi})$ that are much smaller. In applications we estimate $S(\hat{u}, \hat{U}; \hat{\psi})$ by computational approximation of the dual problem.

2.2 The Do-nothing Error Estimate and Indicator

To minimize loss of sharpness, we investigate an approach where the weak form is used directly in a posteriori error estimates, without integration by parts to the strong form, and using Cauchy-Schwarz inequality and interpolation estimates. We here refer to this direct form of a posteriori error representation as the "do-nothing" approach.

In terms of the the exact adjoint solution $\hat{\varphi}$, the output error with respect to a weak solution $\hat{u}$ can be represented as:

$$|M(\hat{u}) - M(\hat{U})| = |((R(\hat{U}), \hat{\varphi}))| = |\sum_{K \in \mathcal{T}_n} ((R(\hat{U}), \hat{\varphi}))_K| \tag{8}$$

This error representation involves no approximation or inequalities. We thus refer to the following error indicator based on the representation as the *do-nothing error indicator*:

$$e_N^K \equiv ((R(\hat{U}), \hat{\varphi}))_K \tag{9}$$

A computable estimate and error indicator are again based on the computed approximation $\hat{\varphi}_h$ of the dual solution:

$$|M(\hat{u}) - M(\hat{U})| \approx |((R(\hat{U}), \hat{\varphi}_h))| \tag{10}$$

$$e_{N,h}^K \equiv ((R(\hat{U}), \hat{\varphi}_h))_K \tag{11}$$

where we may lose reliability of the global error estimate by the Galerkin orthogonality property, which states that the $((R(\hat{U}), \hat{\varphi}_h))$ vanishes for a standard Galerkin finite element method if $\hat{\varphi}_h$ is chosen in the same space as the test functions. Although, in the setting of a stabilised finite element method this is not the case, as we will see below.

2.3 Adaptive Algorithm

We now present an algorithm for adaptive mesh refinement based on the a posteriori output error estimate (8). For simplicity, here we use the same space mesh and the same time step length for all time steps.

Given an initial coarse computational space mesh $\mathcal{T}^0$, start at $k = 0$, then do the following:

1. Compute approximation of the primal problem using $\mathcal{T}^k$.
2. Compute approximation of the dual problem using $\mathcal{T}^k$.
3. If $|\sum_{K\in\mathcal{T}_k}\mathcal{E}^k_{K,h}| < \text{TOL}$ then STOP, else:
4. On the basis of the size of the local error indicator $\mathcal{E}^k_{K,h}$, mark a fixed fraction of the elements in $\mathcal{T}^k$ for refinement. Obtain a new refined mesh $\mathcal{T}^{k+1}$, using a standard algorithm for local mesh refinement.
5. Set $k = k + 1$, then goto (1).

3 Direct Finite Element Simulation

The main elements of Direct Finite Element Simulation of turbulent flow are now presented, in the form of a finite element method with residual based stabilization, quantitative a posteriori error estimation, and an adaptive algorithm.

3.1 The Eulerian cG(1)cG(1) Method

The cG(1)cG(1) method is a variant of G2 using the continuous Galerkin method cG(1) in space and time. With cG(1) in time the trial functions are continuous piecewise linear and the test functions piecewise constant. cG(1) in space corresponds to both test functions and trial functions being continuous piecewise linear. Let $0 = t_0 < t_1 < \ldots < t_N = T$ be a sequence of discrete time steps with associated time intervals $I_n = (t_{n-1}, t_n)$ of length $k_n = t_n - t_{n-1}$ and space-time slabs $S_n = \Omega \times I_n$, and let $W^n \subset H^1(\Omega)$ be a finite element space consisting of continuous piecewise linear functions on a mesh $\mathcal{T}_n = \{K\}$ of mesh size $h_n(x)$ with W^n_w the functions $v \in W^n$ satisfying the Dirichlet boundary condition $v|_{\partial\Omega} = w$.

We now seek functions $\hat{U} = (U, P)$, continuous piecewise linear in space and time, and the cG(1)cG(1) method for the NS equations 01, with homogeneous Dirichlet boundary conditions reads: For $n = 1, \ldots, N$, find $(U^n,\ P^n) \equiv (U(t_n),\ P(t_n))$ with $U^n \in V^n_0 \equiv [W^n_0]^3$ and $P^n \in W^n$, such that

$$\begin{aligned}&((U^n - U^{n-1})k_n^{-1} + \bar{U}^n \cdot \nabla\bar{U}^n, v) + (2\nu\epsilon(\bar{U}^n), \epsilon(v))\\ &\quad -(P^n, \nabla\cdot v) + (\nabla\cdot\bar{U}^n, q) + SD_\delta(\bar{U}^n, P^n; v, q)\\ &= (f, v) \quad \forall (v, q) \in V^n_0 \times W^n \end{aligned} \tag{12}$$

where $\bar{U}^n = 1/2(U^n + U^{n-1})$, with the stabilizing term

$$SD_\delta(\bar{U}^n, P^n; v, q) \equiv (\delta_1(\bar{U}^n \cdot \nabla\bar{U}^n + \nabla P^n - f), \bar{U}^n \cdot\ \nabla v + \nabla q) + (\delta_2 \nabla\cdot\bar{U}^n, \nabla\cdot v)$$

with $\delta_1 = 1/2(k_n^{-2} + |U|^2 h_n^{-2})^{-1/2}$ in the convection dominated case $\nu < \bar{U}^n h_n$ and $\delta_1 = \kappa_1 h_n^2$ otherwise, $\delta_2 = \kappa_2 h_n$ if $\nu < \bar{U}^n h_n$ and $\delta_2 = \kappa_2 h_n^2$ otherwise, with κ_1 and κ_2 positive constants of unit size (here we have $\kappa_1 = \kappa_2 = 1$), and

$$
\begin{aligned}
(v, w) &= \sum_{K \in \mathcal{T}_n} \int_K v \cdot w \, \mathrm{d}x \\
(\epsilon(v), \epsilon(w)) &= \sum_{i,j=1}^{3} (\epsilon_{ij}(v), \epsilon_{ij}(w))
\end{aligned}
$$

We note that the time step k_n is given by the mesh size h_n, with typically

$$
k_n \sim \frac{\min_x h_n(x)}{\bar{U}^n(x)}
$$

4 Parallel-in-time potential for turbulent flow

Our approach to investigating the parallel-in-time potential of computation of mean-value outputs is to study the stability factor $S = S(\hat{u}, \hat{U}; \hat{\psi}) = \|\hat{\varphi}\|_{\hat{V}'}$. In the global norm error estimate (6), S multiplied with the residual bounds the output error, equivalently output sensitivity. In the error representation (8), which the do-nothing adaptive method uses directly, we may also have local cancellation between the residual and stability factor.

We may view the growth of S with regard to the length of the time interval $I = (0, T]$, i.e. $S(T)$ as a measure of the propagation of a given perturbation, or given data as initial condition.

For the heat equation, the model parabolic problem, we expect:

$$
S(T) \leq C \tag{13}
$$

with C a constant, i.e. the perturbation does not propagate, but dissipates away. In this interpretation of a prabolic problem, we may thus say that an initial perturbation does not affect the error after a given time.

For the wave equation, the model hyperbolic problem, we expect:

$$
S(T) \leq CT \tag{14}
$$

[FIXME: check if this is true, should be due to energy conservation, possibly include computations for heat and wave as verification]] i.e., we expect a perturbation to propagate outwards with no dissipation.

4.1 Guessing initial values

In a standard serial time-stepping, solving one time step means computing the initial value of the next step. In a parallel-in-time method however, the initial value of the time steps to be solved are unknown, and need to be guessed or approximated. We can thus view the main challenge of a parallel-in-time method to approximate the initial value of the time steps.

In a parabolic problem with the property (13), the initial value of a time step does not require information very far back in time. This property thus acts to make partitions of the time-interval less dependent on each other.

4.2 The parabolic nature of turbulent flow

By investigating S we may get a measure of how parabolic a problem is, and thus how much potential it has for parallel-in-time scalability.

To assess the parallel-in-time potential of turbulent flow, we apply the DFS methodology to compute $S(T)$ for incompressible Euler for a cube model problem described below, where the mean-value output is drag. In Fig. 6 we plot $S(T)$ for the range $T = [20, 320]$. We observe that $S(T)$ after an initial transient does not grow, it is constant in T.

This behavior appears to indicate that to compute a mean-value output in turbulent flow behaves like a parabolic problem, specifically it has the property (13). Thus we expect large potential for parallel-in-time computation for turbulent flow.

5 Parallel-in-time DFS for turbulent flow

To test if our hypothesis that turbulent flow has large parallel-in-time potential is true, we here design a rudimentary parallel-in-time method based on DFS for the model problem below.

The method is embarassingly parallel, and is based on the assumption that a representative initial value for the mean output can be computed.

5.1 Rudimentary parallel-in-time DFS

1. Partition the time interval I into M partitions $I = \cup I_i$ of equal size K_M.
2. Extend each partition in time by K_M to the left, to allow computation of the initial value, the new extended partitions are denoted $\bar{I}_i$. The extended partitions now have an overlap of K_M.
3. Compute the the solutions $\hat{U}_i$ in all extended partitions $\bar{I}_i$ in parallel.
4. Sum together all solutions restricted to the original partitions $\hat{U}_\Sigma = \sum_{i=0}^{M} \hat{U}_{I_i}$
5. Use $\hat{U}_\Sigma$ to compute the mean-value output: $N(\sigma(\hat{U}_\Sigma))$. Alternatively, compute the local mean-values in parallel and take the global mean value: $\sum_{i=0}^{M} \frac{N(\sigma(\hat{U}_{I_i}))}{M}$

In the model problem section below, we use the method to carry out parallel-in-time computations for turbulent flow around a cube, and verify against serial-in-time reference computations. In Fig.7 we show the global mean-value drag coefficient $N(\sigma(\hat{U}_\Sigma))$, and in Fig.8 we show the drag coefficients over time for two of the partitions.

6 Turbulent Model Problem

The parallel-in-time DFS method is now investigated in a model problem; high Reynolds number flow past a cube in a channel. The high Reynolds number flow is approximated by zero viscosity in the cG(1)cG(1) model, and slip boundary conditions with zero skin friction. Thus there is no viscous dissipation in the model, but the computed (ϵ -weak) solutions exhibit turbulent dissipation in the wake through the residual based stabilization of the method. The mean output quantity is the drag coefficient in a DFS model of the cube model problem. We study the stability factor S, and the convergence of the drag coefficient output quantity.

The model problem is defined by a unit velocity inflow boundary condition in the x-direction past a unit cube located at origo in a pipe in the interval $x = [-10, 30]$ with radius 10, over the time interval I = [0, 400]. The outflow boundary is modelled by a zero pressure boundary condition, and a free slip boundary condition is applied everywhere else. The cube geometry is chosen to minimize the influence of the boundary condition; the cube geometry is exactly represented by the mesh, and the sharp edges trigger flow separation so that boundary layer effects are avoided.

The initial coarse mesh has 3500 vertices, and the do-nothing adaptive method is applied to successively refine the mesh for 14 iterations. A sequence of adaptive meshes is thus generated with ca. 200k vertices in the finest mesh, see Fig. 1. For each mesh we compute the primal and the adjoint solutions, see Fig. 2 and Fig. 3.

We study the stability of the adjoint solution with respect to the length of the time interval $I = (0, T]$ i.e. the stability factor that appears in the global norm a posteriori error estimate (6).

Convergence with respect to the drag coefficient can be seen in Fig. 5, where for the last four adaptive iterations the drag coefficient vary less than 1%.

7 Parallel-in-time DFS computation

We now use the rudinmentary parallel-in-time DFS method to carry out parallel-in-time computations for the turbulent flow cube model problem, and verify against serial-in-time reference computations.

In Fig.7 we show the global mean-value drag coefficient $N(\sigma(\hat{U}_\Sigma))$. The difference in mean-value output between the serial- and parallel-in-time results are ca. 2%, within the error tolerance of the serial-in-time results.

8 Discussion

The development of a PIT methods for turbulent flow is still an open problem, and is a very active field. The development of a scalable PIT method would allow enormously effective computations on the massively parallel supercomputers available today, and the exascale computers available in the near future.

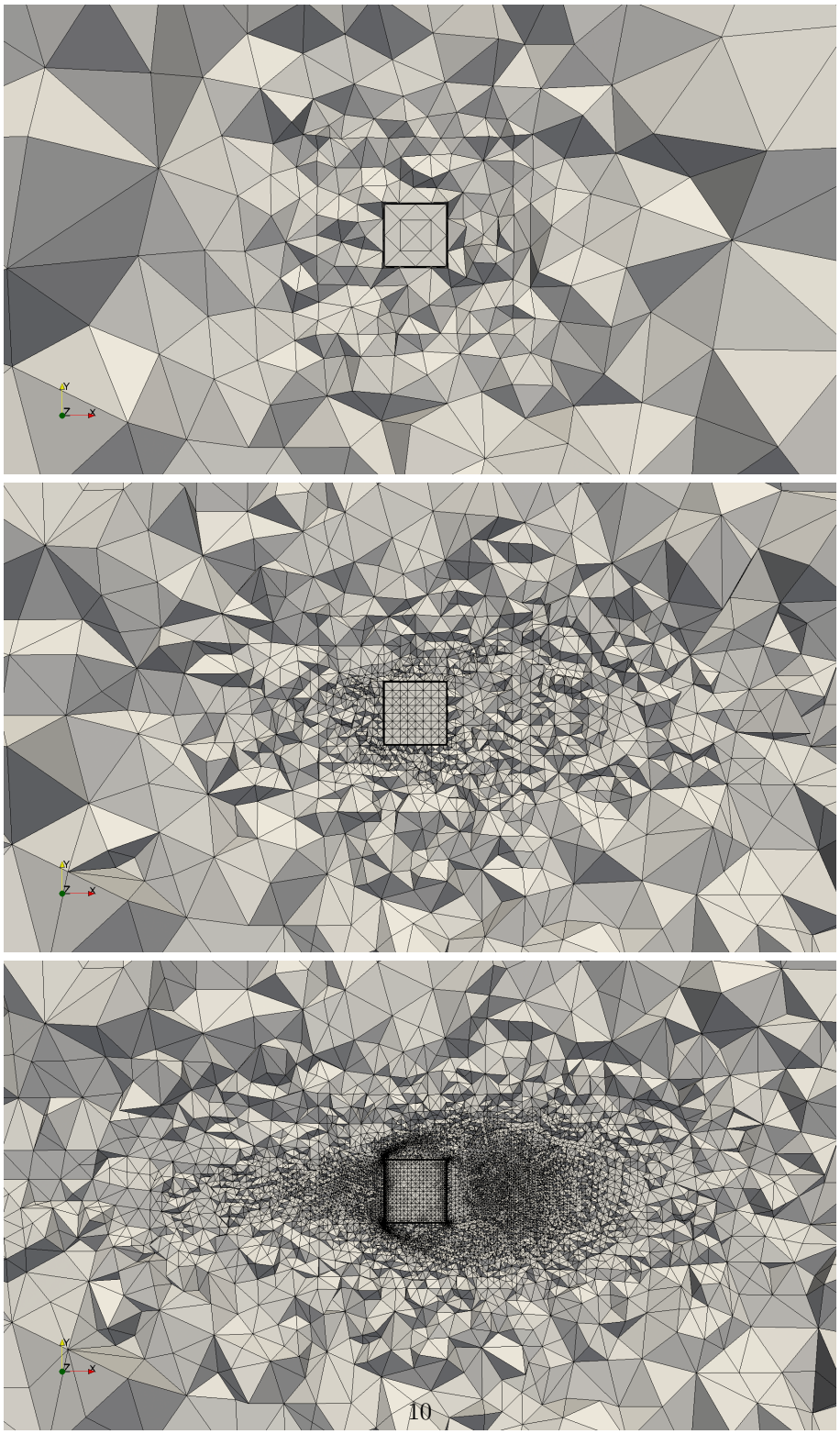

Figure 1: Adaptively refined meshes for iterations 0, 5 and 15.

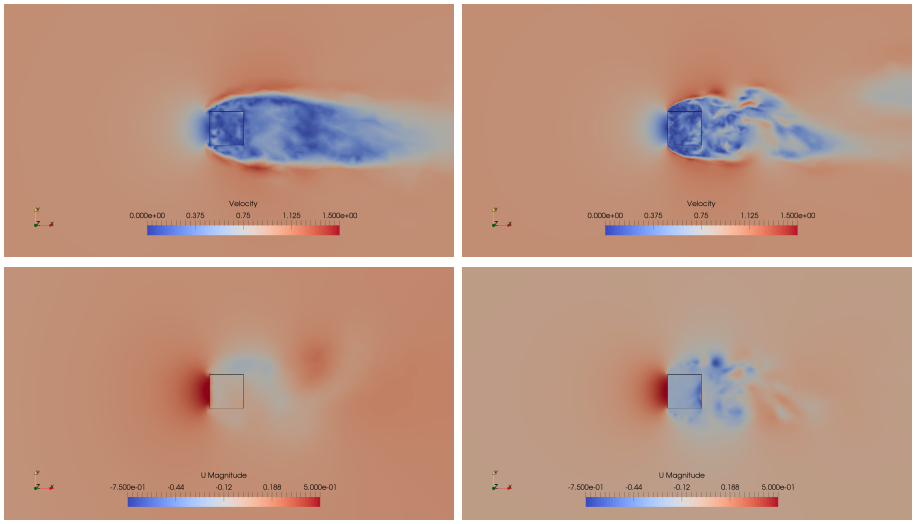

Figure 2: Snapshots of the primal velocity (top) and pressure (bottom) for adaptive iterations 5 and 15.

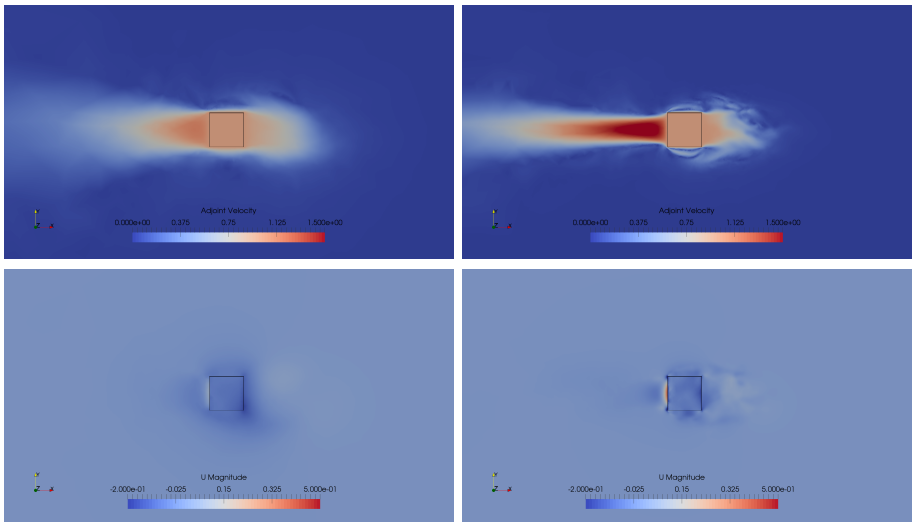

Figure 3: Adjoint velocity φ_h (top) and pressure θ_h (bottom) for adaptive iterations 5 and 15

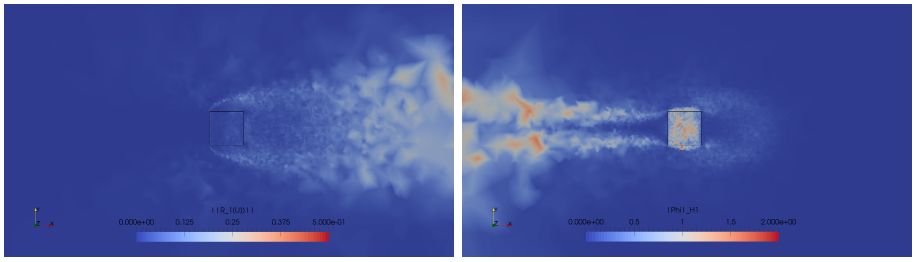

Figure 4: Momentum residual (left) and gradient of the adjoint velocity (right) for adaptive iteration 15.

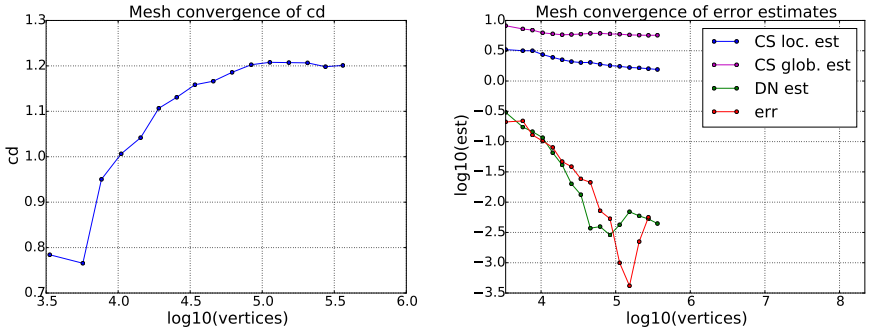

Figure 5: Mesh convergence of the drag coefficient cd (left) and error estimates (right).

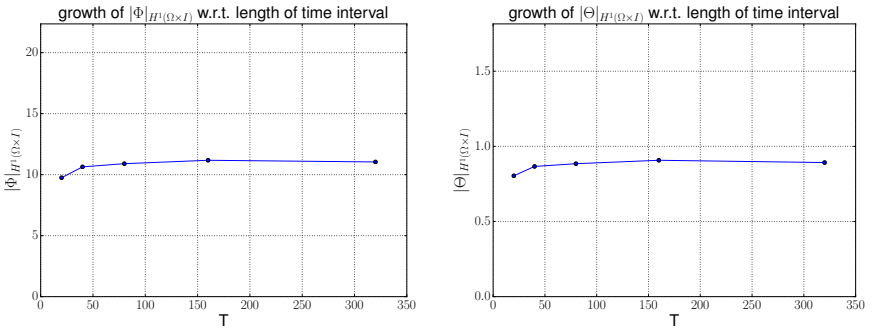

Figure 6: Growth of the stability factors for momentum (left) and continuity (right) with regard to the length of the time interval $I = (0, T]$.

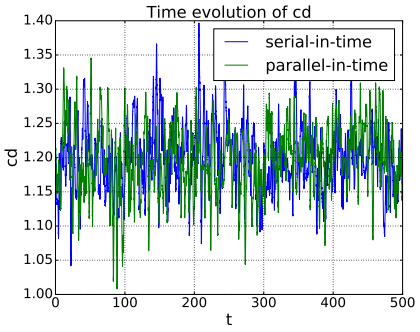

Figure 7: Drag coefficients for the parallel-in-time DFS method, compared to serial-in-time DFS as reference.

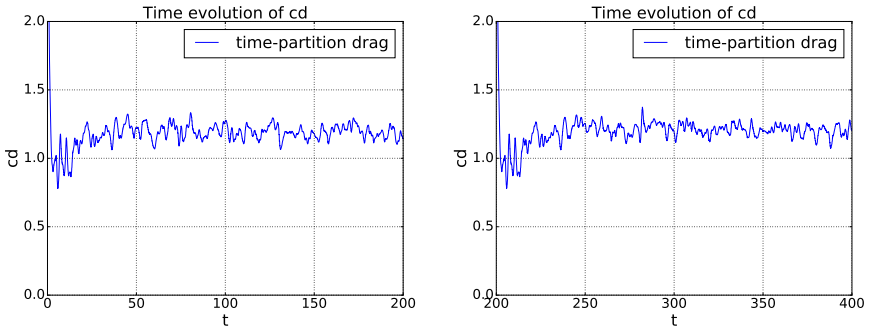

Figure 8: Drag coefficients for the parallel-in-time DFS method for partition I_2 (left) and I_4 (right), note the startup phenomenon in the extended time-interval, which is cut away before computing the mean output.

It would open up real-time or near-real-time computation of grand-challenge problems such as flow around aircraft.

In this paper we have developed a prototype PIT method in the DFS setting. We have tested the method on a cube model problem, and verified that the PIT method gives the same mean-value results as the serial-in-time reference method, within the tolerance of the DFS method.

The verification results show that the method is promising and in future work we plan a more detailed study on complex flow around aircraft, and further development of the method.

References

[1] J.-L. Lions, Yvon Maday, and Gabriel Turinici. A "parareal" in time discretization of PDE's. *Comptes Rendus de l'Acadmie des Sciences - Series I - Mathematics*, 332:661–668, 2001.

[2] L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah. Parallel-in-time molecular-dynamics simulations. *Phys. Rev. E*, 66:057701, Nov 2002.

[3] Guillaume Bal. Parallelization in time of (stochastic) ordinary differential equations. *Math. Meth. Anal. Num.(submitted)*, 2003.

[4] Michael Minion. A hybrid parareal spectral deferred corrections method. *Communications in Applied Mathematics and Computational Science*, 5(2):265–301, 2011.

[5] Paul F. Fischer, Frédéric Hecht, and Yvon Maday. *A Parareal in Time Semi-implicit Approximation of the Navier-Stokes Equations*, pages 433–440. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[6] Roberto Croce, Daniel Ruprecht, and Rolf Krause. *Parallel-in-Space-and-Time Simulation of the Three-Dimensional, Unsteady Navier-Stokes Equations for Incompressible Flow*, pages 13–23. Springer International Publishing, Cham, 2014.

[7] Charbel Farhat and Marion Chandesris. Time-decomposed parallel time-integrators: theory and feasibility studies for uid, structure, and fluid-structure applications. *International Journal for Numerical Methods in Engineering*, 58(9):1397–1434, 2003.

[8] Martin J Gander and Stefan Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM Journal on Scientific Computing*, 29(2):556–578, 2007.

[9] Daniel Ruprecht and Rolf Krause. Explicit parallel-in-time integration of a linear acoustic-advection system. *Computers & Fluids*, 59:72–83, 2012.

[10] Johannes Steiner, Daniel Ruprecht, Robert Speck, and Rolf Krause. Convergence of parareal for the navier-stokes equations depending on the reynolds number. In *Numerical Mathematics and Advanced Applications-ENUMATH 2013*, pages 195–202. Springer, 2015.

[11] José Miguel Reynolds-Barredo, David E Newman, R Sanchez, D Samaddar, Lee A Berry, and Wael R Elwasif. Mechanisms for the convergence of time-parallelized, parareal turbulent plasma simulations. *Journal of Computational Physics*, 231(23):7851–7867, 2012.

[12] Qiqi Wang, Steven A Gomez, Patrick J Blonigan, Alastair L Gregory, and Elizabeth Y Qian. Towards scalable parallel-in-time turbulent flow simulations. *Physics of Fluids*, 25(11):110818, 2013.

[13] J Hoffman, J Jansson, N Jansson, R Vilela De Abreu, and C Johnson. Computability and adaptivity in cfd. encyclopedia of computational mechanics, stein, e., de horz, r. and hughes, tjr eds, 2016.

[14] Jean Leray. Sur le mouvement d'un liquide visqueux emplissant l'espace. *Acta mathematica*, 63(1):193–248, 1934.