On the efficiency and robustness of the adjoint method: Applications in steady and unsteady shape optimization in fluid mechanics

T. Skamagkis[1], A.-S.I. Margetis[2], E.M. Papoutsis-Kiachagias[3], K.C. Giannakoglou[4]

[1] *PhD candidate, e-mail:themistoklissk@mail.ntua.gr*
[2] *PhD candidate, e-mail:amargetis@mail.ntua.gr*
[3] *Researcher, PhD, e-mail:vpapout@mail.ntua.gr*
[4] *Professor, e-mail:kgianna@mail.ntua.gr, web page:http://velos0.ltt.mech.ntua.gr/research/*
Parallel CFD & Optimization Unit
National Technical University of Athens (NTUA)
9, Heroon Polytechniou, NTUA Zografou Campus, 15780, Athens, Greece

Abstract

Two issues regarding the efficiency and robustness of the adjoint method applied to shape optimization problems for steady and unsteady flows are being addressed in the present work. The first one, deals with the convergence difficulties of steady adjoint solvers, for which the Recursive Projection Method (RPM), is implemented to stabilize its iterative solver. The second one addresses the significant storage requirements of the flow solution in the unsteady adjoint method for which a compression strategy is implemented based on the ZFP compression library. Both methods have been implemented cojoined to the latest version of OpenFOAM's *adjointOptimisation* library.

1 Introduction

The recent version of OpenFOAM, v2006, contains an optimization library (*adjointOptimisation*) based on the continuous adjoint method, developed by the Parallel CFD & Optimization Unit (PCOpt) of NTUA. The adjoint method, as a tool for computing the gradient of an objective function in CFD-based shape optimization, has proven to be attractive for use in real-world applications as the cost for computing the gradient is independent of the number of design variables. Next to the (primal) flow problem, an adjoint problem must be solved within each optimization cycle.

Depending on the case, significant difficulties in the numerical solution of the adjoint problem, in either steady or unsteady flows may appear. The system of PDEs is stiff and the adjoint transpose-convection (ATC) term, emerging in the adjoint momentum equations, is often responsible for divergence issues [12]. Widely used "remedies" to this problem are the masking of the ATC or its neglection close to the wall, either totally or selectively on a cell-by-cell basis using an appropriate sensor [8, 12]. Another usual cause of stalling or divergence of the adjoint problem might be the inadequate convergence of the preceding flow solver.

In addition to the above, in the adjoint optimization of unsteady flows, the adjoint equations must be solved backwards in time, requiring the instantaneous primal fields to be available at

each time-step of the adjoint solver; this noticeably increases storage requirements for industrial applications. Various techniques have been proposed to deal with this, including the check-pointing technique [4, 19] or the use of a lower order model [18].

Convergence difficulties of the adjoint equations compromise the robustness of the overall method and getting the optimized solution becomes challenging. Among other, this is the case for solving flow problems where mild unsteadiness is present using a steady solver. Eliminating or smoothing the ATC term, as previousely mentioned, may lead to suboptimal results since erroneous sensitivity derivatives (SDs) might be computed. In this work, the Recursive Projection Method (RPM) [15] is used to alleviate the aforementioned convergence difficulties of the primal and adjoint equations governing incompressible steady flows.

On the other hand, to relax the storage requirements of the unsteady adjoint optimization, a *compressed full storage* strategy is implemented and assessed. To this end, the ZFP compression library [9, 11] is integrated within OpenFOAM as part of the *adjointOptimisation* library.

2 Flow and Adjoint PDEs

Equations governing the steady/unsteady flow of an incompressible fluid (primal problem), in turbulent flows with the Spalart-Allmaras [16] one-equation turbulence model PDE and the Eikonal equation to compute distances Δ from the walls, are:

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \tag{1a}$$

$$R^v_i = \frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} + \frac{\partial p}{\partial x_i} = 0 \ , \qquad i = 1, 2(, 3) \tag{1b}$$

$$R^{\widetilde{\nu}} = \frac{\partial \widetilde{\nu}}{\partial t} + v_j \frac{\partial \widetilde{\nu}}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial \widetilde{\nu}}{\partial x_j}\right] - \frac{c_{b2}}{\sigma}\left(\frac{\partial \widetilde{\nu}}{\partial x_j}\right)^2 - \widetilde{\nu}\, P(\widetilde{\nu}) + \widetilde{\nu}\, D(\widetilde{\nu}) = 0 \tag{1c}$$

$$R^{\Delta} = \frac{\partial}{\partial x_j}\left(\frac{\partial \Delta}{\partial x_j}\Delta\right) - \Delta\frac{\partial^2 \Delta}{\partial x_j^2} - 1 = 0 \tag{1d}$$

where v_i are the velocity components, p is the pressure divided by the fluid density, $\tau_{ij} = (\nu + \nu_t)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)$, ν and ν_t are the bulk and eddy viscocity coefficients respectively and $P(\widetilde{\nu})$ and $D(\widetilde{\nu})$ are the production and dissipation terms, respectively, [20]. The temporal terms $\frac{\partial v_i}{\partial t}$ and $\frac{\partial \widetilde{\nu}}{\partial t}$, in eq. 1b and 1c respectively, are ommited for steady-state flows. Spalding's law of the wall is used in order to model the near wall behavior of the flow [17]. Coefficients in eq. 1c can be found in [20].

The adjoint boundary conditions are omitted in the interest of space, since they also depend on the flow boundary conditions (omitted too) and the objective function J of the problem under consideration. In the present studies, all objective functions consist of boundary integrals and do

not, thus, affect the field adjoint equations, which are [13]

$$R^q = -\frac{\partial u_j}{\partial x_j} = 0 \tag{2a}$$

$$R^u_i = -\frac{\partial u_i}{\partial t} + u_j \frac{\partial v_j}{\partial x_i} - \frac{\partial \left(v_j u_i\right)}{\partial x_j} - \frac{\partial \tau^\alpha_{ij}}{\partial x_j} + \frac{\partial q}{\partial x_i} + \widetilde{\nu_a} \frac{\partial \widetilde{\nu}}{\partial x_i} - \frac{\partial}{\partial x_l}\left(\widetilde{\nu_a}\widetilde{\nu}\frac{C_Y}{Y} e_{mjk} \frac{\partial v_k}{\partial x_j} e_{mli}\right) = 0 \quad i = 1, 2(, 3) \tag{2b}$$

$$R^{\widetilde{\nu_a}} = -\frac{\partial \widetilde{\nu_a}}{\partial t} - \frac{\partial \left(v_j \widetilde{\nu_a}\right)}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial \widetilde{\nu_a}}{\partial x_j}\right] + \frac{1}{\sigma}\frac{\partial \widetilde{\nu_a}}{\partial x_j}\frac{\partial \widetilde{\nu}}{\partial x_j} + 2\frac{c_{b2}}{\sigma}\frac{\partial}{\partial x_j}\left(\widetilde{\nu_a}\frac{\partial \widetilde{\nu}}{\partial x_j}\right) + \widetilde{\nu_a}\widetilde{\nu}C_{\widetilde{\nu}} + \frac{\partial \nu_t}{\partial \widetilde{\nu}}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + (-P + D)\widetilde{\nu_a} = 0 \tag{2c}$$

$$R^{\Delta_\alpha} = -2\frac{\partial}{\partial x_j}\left(\Delta_\alpha \frac{\partial \Delta}{\partial x_j}\right) + \widetilde{\nu}\widetilde{\nu_a}C_\Delta = 0 \tag{2d}$$

where u_i are the adjoint velocity components, q the adjoint pressure, $\widetilde{\nu_a}$ the adjoint to the turbulence model variable, $\tau^\alpha_{ij} = (\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$ are the adjoint stresses and Δ_α the adjoint distance from the walls. Terms $C_{\widetilde{\nu}}$, C_Y and C_Δ can be found in [20]. The temporal terms $\frac{\partial u_i}{\partial t}$ and $\frac{\partial \widetilde{\nu_a}}{\partial t}$ in eq. 2b and 2c respectively are ommited for steady-state flows. The term $u_j \frac{v_j}{x_i}$ is the so-called ATC term, the main culprit responsible for the divergence of the adjoint equations. The adjoint system includes the differentiated Spalding's law, as described in [13]. Both sets of equations are solved numerically using the OpenFOAM library which makes use of a cell-centered finite volume scheme. The SIMPLE and PISO algorithms were used to solve the steady and unsteady flow equations, respectively. In all cases demonstrated within this work, the surfaces to be optimized and the grid around them were parameterized with volumetric B-splines, the coordinates of the control points of which stand for the design variables $\mathbf{b}$.

3 The Recursive Projection Method (RPM)

The RPM is a stabilization method developed for the purpose of treating the convergence difficulties of a fixed-point iteration scheme $\boldsymbol{U}^{(n+1)} = F(\boldsymbol{U}^{(n)})$, where $\boldsymbol{U}$ is the array of (primal or adjoint) unknowns, n the iteration counter and $F : \mathbb{R}^N \to \mathbb{R}^N$. The convergence of the scheme is determined by the magnitudes of the eigenvalues of the Jacobian matrix F_U. If all of them are less than unity, the iterative scheme converges to a fixed-point solution. If one, or more, eigenvalues are greater than unity, the scheme diverges. The RPM may assist in getting a converged solution, provided that the number of eigenvalues that are larger than one (i.e. the dominant eigenvalues) is relatively low, compared to the size of the system at hand. First, the method needs to approximate the eigenvectors associated with the dominant eigenvalues, to be referred to as the diverging modes. Once these are known, and suppose that they are m in total, a basis $Z \in \mathbb{R}^{N \times m}$ is formed, spanned by these m eigenvectors and two subspaces of $\mathbb{R}^N$ are defined: the unstable subspace $\mathbb{P}$ and its orthogonal complement $\mathbb{Q}$. Within the RPM algorithm, this basis is formed incrementally throughout the solution of the primal/adjoint equations. Projection matrices, from $\mathbb{R}^N$ onto these subspaces, using the basis Z are defined as $P = ZZ^T$ and $Q = I - ZZ^T$ and the solution $\boldsymbol{U}$ is decomposed into $\mathbf{p} = P\boldsymbol{U}$ and $\mathbf{q} = Q\boldsymbol{U}$, namely the unstable and stable parts respectively.

The core idea of the RPM is to use a Newton-Raphson method to solve for $\mathbf{p}$ while retaining the standard solver for $\mathbf{q}$. More details about its implementation and the method for constucting Z in an efficient manner can be found in [15, 3]. The RPM has been developed as a shell code

around the solvers of the *adjointOptimisation* library of OpenFOAM. Depending on the case, the RPM could be applied to the two sets of equations, primal and adjoint, excluding eqs. 1d and 2d which are solved decoupled from the rest.

4 Data Compression in Unsteady Adjoint Optimization

For the adjoint optimization of unsteady flows, the unsteady adjoint equations must be solved backwards in time. The storage of the primal solution at all time-steps (*full storage* strategy) can become memory intensive in real-world applications. The check-pointing technique [4, 19] may efficiently relax the memory requirements of the adjoint computation, but at the expense of computational overhead. This paper proposes a *compressed full storage* strategy, in which, at all time-steps, the instantaneous primal fields are compressed (in a lossy manner) using the ZFP [9, 10] compression package and then appended to the database. One of the purposes of this study is to assess the lossy compression offset between data reduction and compression accuracy.

ZFP is a state-of-the-art compression package, developed in C/C++, to compress integer and floating-point data stored in d-dimensional arrays, with $d \in [1, 4]$. These arrays are partitioned into independent blocks of 4^d values each. In each block, the data values are converted to a block-floating-point representation and the resulting 63-bit signed integer values are decorrelated using a near orthogonal transform. The transform coefficients are reordered in roughly monotonically decreasing order and, then, losslessly compressed using embedded coding. A thorough analysis of the ZFP algorithm can be found in [9, 10].

In this study, the data are compressed as 2D arrays using the *fixed-precision* mode of ZFP. To do so, each data list of size N is (optionally) reordered using the *ZFP-sorting* technique, then transformed into a $4 \times m$ matrix, where $m = \lceil N/4 \rceil$ and, finally, compressed. The primal fields that must be stored are the cell-centered p, v_i, the fluxes ϕ at the cell-faces and, for turbulent flows $\widetilde{\nu}$. ν_t can be re-computed based on the decompressed $\widetilde{\nu}$ using an algebraic expression. In parallel runs, p, v_i and $\widetilde{\nu}$ along the subdomain interfaces are not stored and the corresponding values are restored upon decompression based on the neighbouring cell-center values of the internal domain, whereas in 2D cases, only the values corresponding to the two solution directions are stored. Each scalar list, that holds the scalar or vector primal field values at the internal and the domain boundary, is compressed independently and stored as binary data [6]. Hereafter, each of these lists will be referred to simply as the 'list'. In view of the contradicting objectives of data reduction and compression accuracy and since ZFP is designed for structured data whose values vary reasonably smoothly, the *ZFP-sorting* variant in which the data are sorted in ascending or descending order before compressed is tried first. Since the rearrangement is value-based, in each time-step, each field must be sorted anew, increasing potentially the computational cost and the storage requirements since extra indexing lists (different for each list) must also be stored. As a middle ground solution, the same indexing lists are retained for a number of consecutive time-steps, exploiting the fact that, to a large extent, the ordering of the sorted data stream is similar between adjacent time-steps.

To evaluate the reduction in memory size, two compression ratio metrics are considered. The *overall compression ratio* CR_0 is defined as the ratio of the memory size for storing all the primal fields to be compressed and that of the compressed streams including the indexing lists. CR_0 reflects the total reduction in memory footprint and helps concluding whether an unsteady adjoint simulation fits into the available system memory or not. On the other hand, the *ZFP compression ratio* CR_{ZFP} is the ratio of the memory size of all the primal fields to be compressed and that of the compressed streams, reflecting the real performance of the ZFP algorithm. SDs ($\delta J/\delta \mathbf{b}$) computed when the primal solution is stored uncompressed will be referred to as "*SDs without compression*" and those $\left(\delta J^{'}/\delta \mathbf{b}\right)$ computed using the lossy compressed primal solution as "*ZFP*

SDs". To assess them, the *peak-signal-to-noise-ratio* ($PSNR$ or Q) [9, 5, 1, 7] is used. For a discrete uncompressed signal ϕ, standing for an array of N floating-point numbers, and its approximation $\phi^{'}$, Q reads

$$Q\left(\phi,\phi^{'}\right)=10\log_{10}\frac{\left(\phi_{max}-\phi_{min}\right)^{2}}{\frac{1}{N}\sum_{i=1}^{N}\left(\phi_{i}-\phi_{i}^{'}\right)^{2}} \tag{3}$$

The higher the value of the $PSNR$, the lower the overall error of the second dataset, indicating compression of higher quality. Next to Q, the angle θ between $\delta J/\delta\mathbf{b}$ and $\delta J^{'}/\delta\mathbf{b}$ vectors in the multidimensional space and the difference $\Delta\left(\delta J/\delta\mathbf{b}\right)$

$$\theta=cos^{-1}\frac{\frac{\delta J}{\delta\mathbf{b}}\cdot\frac{\delta J^{'}}{\delta\mathbf{b}}}{\left\|\frac{\delta J}{\delta\mathbf{b}}\right\|\left\|\frac{\delta J^{'}}{\delta\mathbf{b}}\right\|}\ ,\quad\Delta\left(\frac{\delta J}{\delta\mathbf{b}}\right)=\frac{\left\|\frac{\delta J}{\delta\mathbf{b}}-\frac{\delta J^{'}}{\delta\mathbf{b}}\right\|}{\left\|\frac{\delta J}{\delta\mathbf{b}}\right\|} \tag{4}$$

are also computed.

5 Shape Optimization of a Cylinder in a Periodic Flow

The flow around a circular cylinder is periodic beyond $Re\approx 47$ [2]. Thus, when attempting to optimize the initially cylindrical shape, for any objective function J, it would be necessary to use an unsteady simulation of the periodic flow and integrate the objective function in time. Next to this, an unsteady adjoint formulation would have been necessary. Considering that the selected objective is to reduce the (mean) drag exerted on the surface, the optimization is expected to minimize the frontal area of the body, squeezing it in the transversal direction as a means to suppress the vortex shedding. By doing so, after a certain point, the suppresion of vortices would lead to a steady flow around the optimized shape. Therefore, one might start the optimization with a steady solver and treat any convergence difficulties that may occur (using the RPM in this study).

In the selected demo case, $Re=90$ and a grid with 12880 cells was used. Although the flow was solved as purely 2D, OpenFOAM uses a 3D grid with one cell in the spanwise direction and the volumetric B-splines control lattice consisted of $11\times 10\times 3$ CPs. The objective function J is the time-averaged (averaged over a period, in case of vortex shedding) drag exerted on the solid wall. An equality constraint on the body area/volume (to be kept equal to its initial value) was imposed using the projection technique of [14].

When solving the primal equations, on the baseline geometry, using a steady solver and without the RPM, the residuals ended up in limit cycle oscillations. The RPM made them converge adequately and the so-computed primal fields were used to solve the adjoint equations. When the RPM was not used for the adjoint equations, these failed to converge. Before using the RPM on the adjoint solver, a couple of simple, yet widely used, "tricks" were also tried. The ATC term was explicitly zeroed out on the cylinder patch, however, the adjoint equations could not yet converge. It was then cancelled completely, in which case the adjoint equations converged. Finally, the RPM was used to converge the adjoint equations, without arbitrarily altering the adjoint equations. Since the complete set of adjoint equations is solved herein, these SDs are considered to be the correct ones and are compared with finite differences (FD) in fig. 1.

The shape optimization loop discussed below used the RPM during the numerical solution of both the steady primal and adjoint equations. Apart from the first optimization cycle, both sets of equations could converge without "assistance', however, the RPM remained active and accelerated convergence by identifying slowly-decaying modes and adding them to Z. The convergence of both sets of equations can be seen in fig. 2. 15 cycles were performed in total before reaching a local

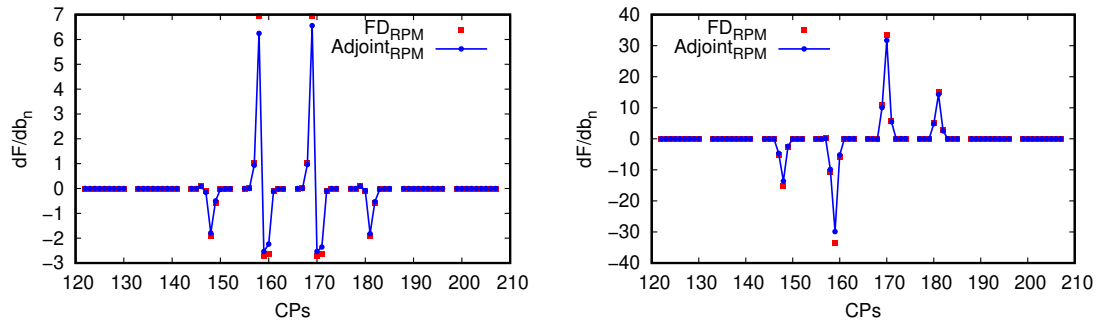


Figure 1: 2D flow around a cylinder. Computed SDs w.r.t. the x (a) and y (b) coordinates of the CPs using the adjoint method (blue) are compared against reference FD (red). The RPM was used to stabilize both the primal and adjoint solver in the case of adjoint and to make the primal equations converge when computing SDs with FD for every movement of the CPs.

minimum and, thereafter, no further reduction was possible. Streamlines and the pressure and velocity magnitude fields, computed with the baseline and optimized shapes are shown in fig. 3.

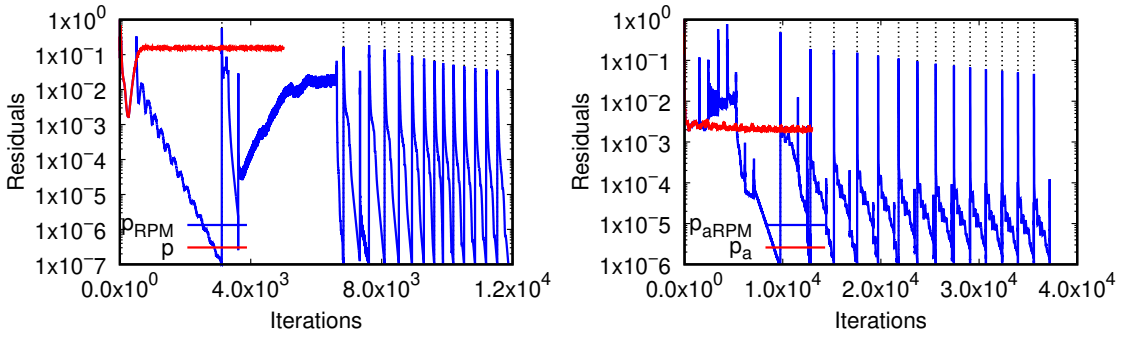


Figure 2: 2D flow around a cylinder. Residuals of the primal (left) and adjoint (right) pressure equations. Red color indicates residuals without implementing the RPM for the primal solution (left) and, using the RPM on the primal but not the adjoint equations (right). Vertical black lines distinguish between successive optimization cycles. On the second optimization cycle, the primal equations initially converge, then the RPM makes a bad approximation to the unstable subspace leading to divergence and, after a number of iterations, the basis is augmented again leading to convergence again.

The flows around the initial and optimized geometries were recomputed using an unsteady solver, with the PISO algorithm for which a constant time-step of $\Delta t = 10^{-3}s$ was used. Since the flow period around the initial geometry (cylinder) is $T \sim 0.68s$, ~ 680 time-steps per period were performed. On the initial geometry, the RPM-assisted steady solver computed $J = 1.17$ and the unsteady solver a mean $\overline{J} = 1.41$. For the shape produced by the steady optimization, $J = 0.769$, whereas its unsteady analysis gave $\overline{J} = 0.771$, clearly indicating that the flow around the optimized shape had become steady.

Finally, the shape optimization was performed anew, this time using an unsteady adjoint. The evolution of the J value is plotted in fig. 4a. Although the steady and unsteady optimization runs

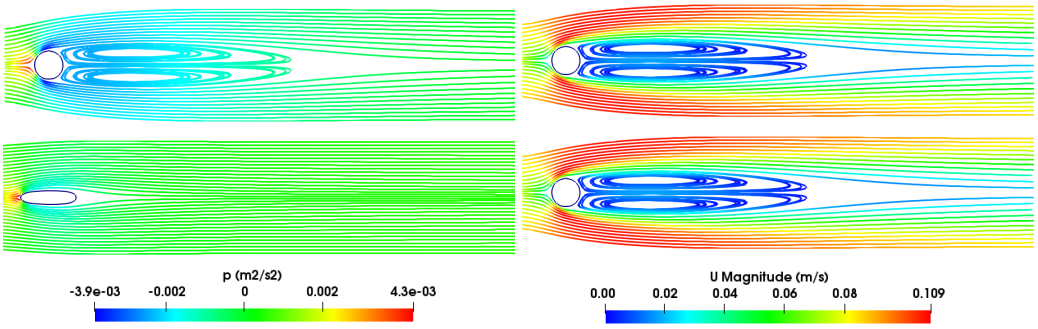
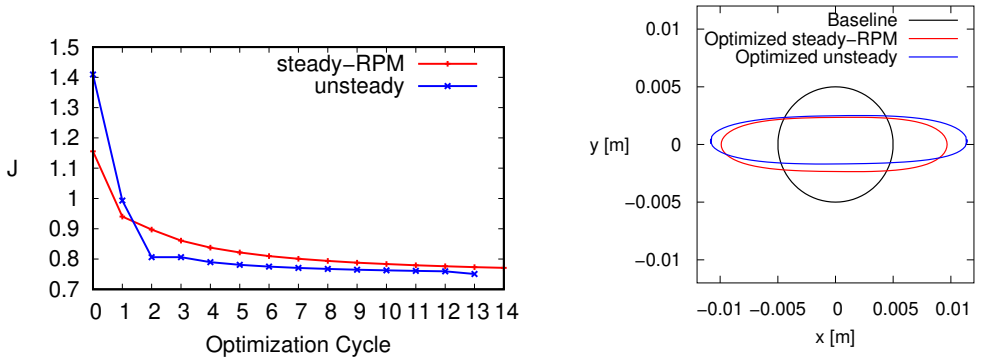


Figure 3: 2D flow around a cylinder. Streamlines around the baseline and optimized geometries, coloured based on the pressure (left) and velocity magnitude fields (right). The optimization eliminated the vortices behind the body and reduced the adverse pressure gradient.

do not lead to identical shapes, fig. 4b, the drag is practically the same.



(a) (b)

Figure 4: 2D flow around a cylinder. (a) Descent of J using a steady solver assisted by the RPM (red) and an unsteady one (blue). (b) Comparison between the initial and optimized cylinder shapes produced by the steady and unsteady optimizations.

To reduce the memory footprint of the unsteady adjoint, the *compressed full storage* strategy was afterwards implemented. Based on table 1, the use of lossless compression resulted to a small compression ratio ($CR_0 = 1.08$), whereas the additional cost due to the fields' compression and decompression was ~ 1.2 times the cost of the optimization cycle when the data were stored uncompressed. On the other hand, the lossy compression reduced the data size by more than one order of magnitude, increasing the cost by only $\sim 35\%$. The indexing lists were updated every 50 time-steps, their compression ratio was equal to ~ 1.7 and their total compressed size corresponds to 6 to 11% of the total compressed data in memory.

The correlation of the user-defined accuracy in bits with the *overall compression ratio* and the PSNR of the SDs is shown in fig. 5, whereas the uncompressed and absolute error of the reconstructed v_i magnitude is shown in fig. 6. The convergence of the objective function and the optimal shapes after 14 optimization cycles are shown in fig. 7, whereas the *SDs without*

Table 1: 2D flow around a cylinder, CR_0 and CR_{ZFP}, uncompressed and compressed size of the primal solution at the first optimization cycle. The indexing lists were updated every 50 time-steps, their CR was equal to $\sim$1.7 and their total compressed size corresponds to 6 to 11% of the total compressed data in memory.

Compression	***CPU cost***	***Precision*** $\mathbf{p}/\phi/\mathbf{v_i}$ ***(bits)***	$\mathbf{CR_0}$	$\mathbf{CR_{ZFP}}$	***Uncomp. /Comp. size (MB)***	***SDs***			
						Q(dB)	θ**(deg)**	$\mathbf{\Delta}\left(\frac{\delta \mathbf{J}}{\delta \mathbf{b}}\right)$	***Wrong signs /total***
no	100%								
lossless	215.9%		1.09						
lossy	136.5%	12/12/12	8.59	9.14	1092/127.1	72.14	0.24	0.43%	3/144
		10/10/10	11.40	12.38	1092/95.8	59.00	1.07	1.94%	13/144
		08/08/08	16.14	18.19	1092/67.7	38.09	12.43	21.53%	28/144

compression and the *ZFP SDs* of the first cycle are shown in fig. 8.

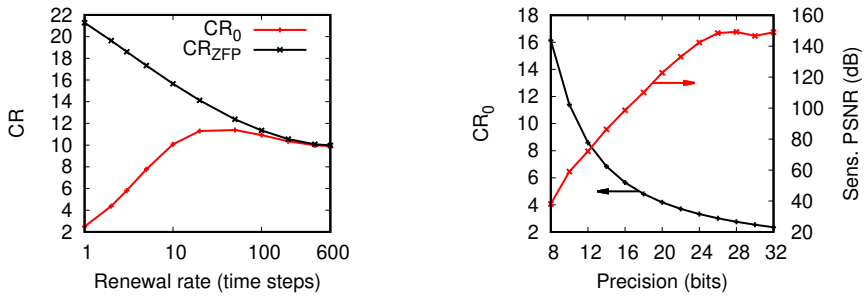


Figure 5: 2D flow around a cylinder. Left: CR_0 and CR_{ZFP} w.r.t. to the indexing lists' renewal rate using 10 bits of precision for all fields; right: CR_0 and the PSNR of the SDs at the first optimization cycle w.r.t. the compression's precision in bits. The indexing lists were updated every 50 time-steps.

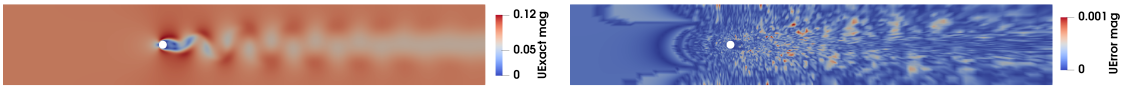


Figure 6: 2D flow around a cylinder. Left: uncompressed $\|v_i\|$; right: absolute error (reconstructed - uncompressed) of $\|v_i\|$ at $t=T/2$ of the first optimization cycle. The PSNR value of the v_i field was equal to $51.52dB$. Compression performed using 10 bits of precision for all fields.

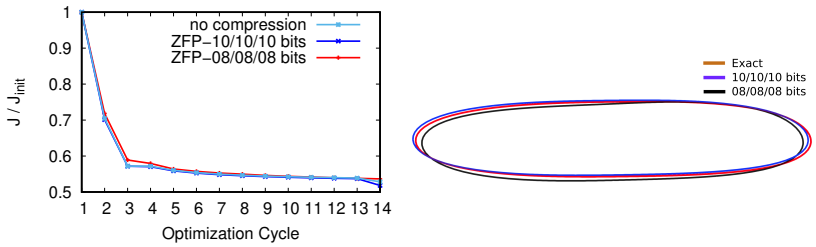
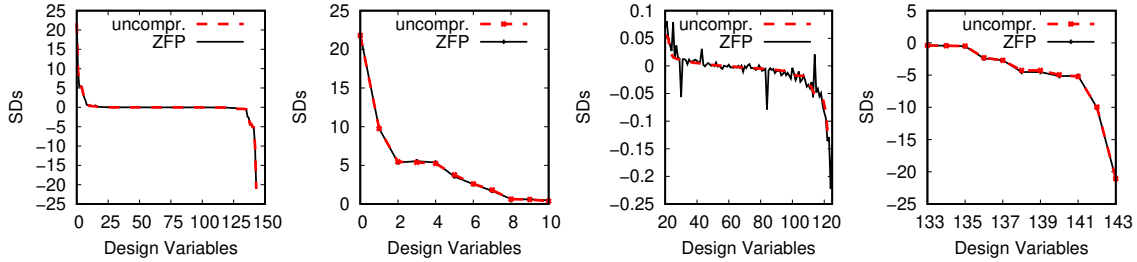


Figure 7: 2D flow around a cylinder. Left: convergence of the J; right: optimized shapes. J was reduced by $\sim$47%. Both the objective function convergence and the optimal shapes were almost unaffected by the lossy compression.



(a) All SDs (b) Close-up view (c) Close-up view (d) Close-up view

Figure 8: 2D flow around a cylinder, *SDs without compression* at the first optimization cycle, sorted in descending order for visualization purposes, paired with the corresponding *ZFP SDs*. Zero-valued SDs are omitted. Compression performed using 10 bits of precision for all fields. Plots 8b, 8c and 8d are close up views of 8a. The PSNR of the SDs is equal to 59.0 dB and $\Delta(\delta J/\delta \mathbf{b}) = 1.94\%$.

6 Closure - Other Applications

In section 5, shape optimization was performed in a typical flow problem with the goal of demostrating the implementation of the RPM (as a stabilization and, occassionally, acceleration method) and a *compressed full storage* strategy based on the *ZFP-sorting* variant of the ZFP compression package to reduce the memory requirements of the backwards in time integrated unsteady adjoint. The RPM was used to overcome the divergence of the adjoint equations in a case with an unsteady flow. Both the primal and adjoint equations were converged using the RPM and the results of the optimization were compared with the solutions provided by an unsteady solver. In a case in which the optimization is expected to suppress the flow unsteadiness, it is reasonable to use a steady solver and numerically treat any convergence difficulties even though the original flow problem would have required an unsteady solver. The *compressed full storage* strategy was implemented both in its lossless mode, which proved non profitable to support the unsteady adjoint, and in its error-bounded lossy mode, which successfully reduced the memory requirements of the unsteady adjoint by more than one order of magnitude at a small computational overhead, while the compression error does not affect the outcome of the optimization.

For the sake of completeness, two more cases, which are much closer to real-world applications, are presented (fig. 9 and fig. 10) and discussed in the corresponding captions.

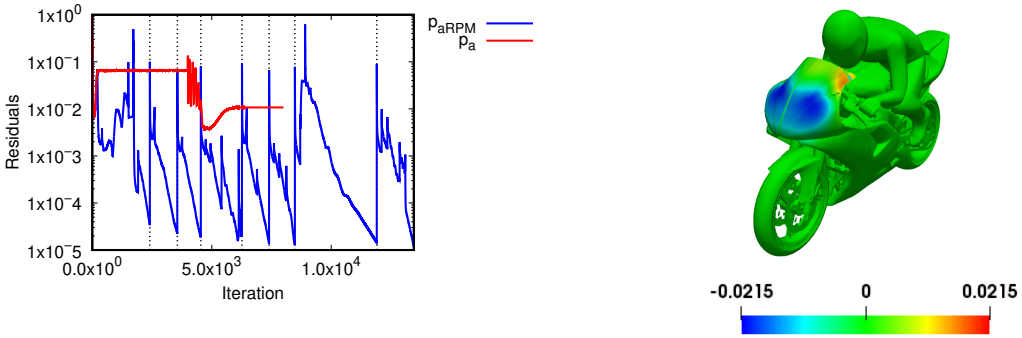


Figure 9: Motorbike tutorial case of the OpenFOAM library. The goal of the optimization is to minimize the whole drag by controlling the shape of the motorbike's fairing. The 3D grid consisted of approximately 10^6 cells and the shape was parameterized with volumetric B-Splines using a control lattice of $7 \times 7 \times 7$ CPs, 216 of which are active, yielding a total of $216 \times 3 = 648$ design variables. The turbulent flow and exhibited unsteadiness so that the steady-state flow solver was unable to converge leading to stagnated, oscillating residuals. In this study, a less intrusive treatment to the ATC term was used, compared to the one used in the tutorial and, consequently, the adjoint equations diverged. The RPM was used to stabilize the adjoint solver and its effect can be seen on the left where, after its activation, the adjoint equations converged. The divergence on the baseline geometry is due to 6 eigenvalues residing outside the unit circle. In all subsequent optimization cycles, the use of the RPM revealed eigenvalues with moduli greater than unity causing divergence of the adjoint equations in all subsequent optimization cycles, rendering its use mandatory. The cumulative normal displacement field computed on the final geometry is plotted on the right. Red/blue color indicates the points on the fairing surface that need to be pushed outwards/ pulled inwards in order for the drag to decrease. Given that the primal equations did not fully converge and, without the RPM, the adjoint solver diverged, the use of the RPM allowed for the optimization to continue.

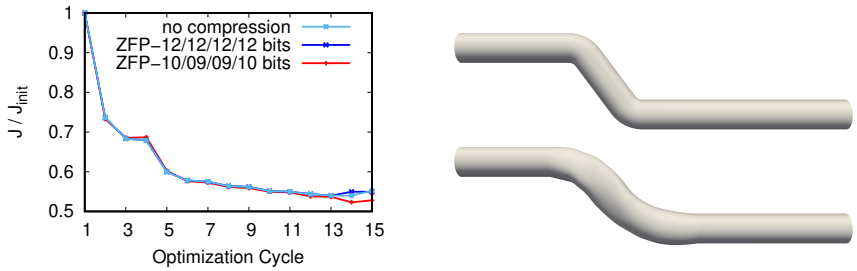


Figure 10: 3D S-bend, unsteady adjoint optimization. The unsteady flow is due to a periodic inlet boundary condition for v_i with $\overline{Re} = \sim 8500$ and $\sim 10^4$ (constant) time-steps per period. The grid having ~ 92000 hexahedra was parameterized using volumetric B-Splines with 360 active CPs. Target of the optimization was the minimization of the total pressure losses between the inlet and outlet boundaries of the fluid domain. Left: J was reduced by $\sim 46\%$ after 13 optimization cycles and its convergence was almost unaffected by the lossy compression error; right: initial (top) and optimized (bottom) shape of the duct.

Table 2: 3D S-bend: for the lossy compression, the indexing lists were updated every 50 time-steps for the first case and every 100 for the second one; their CR was equal to ~ 1.6 and their total compressed size corresponds to 8 to 10% of the total compressed data in memory. The *ZFP SDs* were evaluated at the first optimization cycle. A maximum $CR_0 = 25.56$ was achieved without affecting the outcome of the optimization.

Compression	***CPU cost***	***Precision*** $\mathbf{p}/\phi/\mathbf{v_i}/\widetilde{\nu}$ ***(bits)***	$\mathbf{CR_0}$	***Uncomp. /Comp. size (GB)***	***SDs***			
					Q(dB)	θ**(deg)**	$\mathbf{\Delta}(\frac{\delta \mathbf{J}}{\delta \mathbf{b}})$	***Wrong signs /total***
no	100%							
lossless	149%		1.12					
lossy	136%	12/12/12/12	15.99	190.7/11.9	84.14	0.07	0.14%	2/1080
		10/09/09/10	25.56	190.7/7.46	59.54	1.26	2.24%	39/1080

The RPM was used to deal primarily with the convergence difficulties of the adjoint equations due to the ATC term. Shape optimization of a motorbike's fairing fig. 9 for drag minimization was performed. In this case the primal equations could not fully converge and the adjoint solver diverged. Even though the RPM could not overcome the convergence difficulties of the primal solver it was successfull in stabilizing the adjoint one and the optimization loop, overall. Overall, the RPM has been succesful in stabilizing the steady adjoint solver, provided that the diverging modes are small in number. Indeed, in both cases discussed herein, divergence occured due to the appearance of 2 to 6 diverging modes.

The *compressed full storage* strategy, based on the ZFP algorithm, was implemented on a 3D S-bend duct, showing that lossless compression is expensive and ineffective in data reduction, in contrast to the lossy compression, which can achieve a data reduction by more than an order of magnitude at a small/affordable computational overhead ($\sim 35\%$), table 2. The achieved compression ratios ranging between 16 and 25, lead to acceptable errors in the SDs values, so that the convergence of the objective function and the optimal shapes were almost unaffected by the compression error, fig. 10.

Ackowledgements

Acknowledgements are due to Bayerische Motoren Werke (BMW) for funding parts of the corresponding research.

References

[1] S. Di and F. Cappello. Optimization of Error-Bounded Lossy Compression for Hard-to-Compress HPC Data. *IEEE Transactions on Parallel and Distributed Systems*, 29(1):129–143, 2018.

[2] F. Gianetti and P. Luchini. Structural sensitivity of the first instability of the cylinder wake. *Journal of Fluid Mechanics*, 581:167–197, 2007.

[3] S. Gortz and J. Moller. Evaluation of the recursive projection method for efficient unsteady turbulent CFD simulations. *24th International Congress of the Aeronautical Sciences*, 2004.

[4] A. Griewank and A. Walther. Algorithm 799: Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software*, 26(1):19–45, 2000.

[5] P. Hadjidoukas and F. Wermelinger. A Parallel Data Compression Framework for Large Scale 3D Scientific Data. *ArXiv*, abs/1903.07761, 2019.

[6] S. Hamilton, R. Burns, C. Meneveau, P. Johnson, P. Lindstrom, J. Patchett, and A.S. Szalay. Extreme Event Analysis in Next Generation Simulation Architectures. *Lecture Notes in Computer Science High Performance Computing*, 10266:277–293, 2017.

[7] J. Iverson, C. Kamath, and G. Karypis. Fast and Effective Lossy Compression Algorithms for Scientific Datasets. *Euro-Par 2012 Parallel Processing Lecture Notes in Computer Science*, 7484:843–856, 2012.

[8] G. Karpouzas, E.M. Papoutsis-Kiachagias, T. Schumacher, E. de Villiers, K.C. Giannakoglou, and C. Othmer. Adjoint optimization for vehicle external aerodynamics. *International Journal of Automotive Engineering*, 7(1):1–7, 2016.

[9] P. Lindstrom. Fixed-Rate Compressed Floating-Point Arrays. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2674–2683, 2014.

[10] P. Lindstrom, M. Salasoo, M. Larsen, and S. Herbein. *ZFP Documentation - Release 0.5.5.* Springer, 2019.

[11] LLNL/ZFP. https://github.com/LLNL/zfp, 2020.

[12] C. Othmer. Adjoint methods for car aerodynamics. *Journal of Mathematics in Industry*, 4(6), 2014.

[13] E.M. Papoutsis-Kiachagias and K.C. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications. *Archives in Computational Methods in Engineering*, 32(2):255–299, 2016.

[14] J.B. Rosen. The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960.

[15] G.M. Shroff and H. Keller. Stabilization of unstable procedures: The recursive projection method. *SIAM Journal of Numerical Analysis*, 30(4):1099–1120, 1993.

[16] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. AIAA Paper 1992-439, 30th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, USA, January 6–9 1992.

[17] D. B. Spalding. A single formula for the law of the wall. *Journal of Applied Mechanics*, 28:455–457, 1961.

[18] C. Vezyris, E.M. Papoutsis-Kiachagias, and K.C. Giannakoglou. On the incremental singular value decomposition method to support unsteady adjoint based optimization. *Numerical Methods in Fluids*, 91(7):315–331, 2019.

[19] A. Walther and A. Griewank. Advantages of Binomial Checkpointing for Memory-reduced Adjoint Calculations. *Numerical Mathematics and Advanced Applications*, 2004 pp. (834-843).

[20] A.S. Zymaris, D.I. Papadimitriou, K.C. Giannakoglou, and C. Othmer. Continuous adjoint approach to the Spalart-Allmaras turbulence model for incompressible flows. *Computers & Fluids*, 38(8):1528–1538, 2009.