On the Incremental Singular Value Decomposition Method to Support Unsteady Adjoint-Based Optimization

Christos Vezyris*, Evangelos Papoutsis-Kiachagias, Kyriakos Giannakoglou
School of Mechanical Engineering, Parallel CFD & Optimization Unit (PCOpt)
National Technical University of Athens, Athens, Greece

Correspondence: Christos Vezyris, cvezyris@gmail.com

September 15, 2019

Abstract

[Summary]This paper proposes and evaluates an approximation model based on an incremental Singular Value Decomposition (iSVD) algorithm, for unsteady flow field reconstructions, needed for integrating the unsteady adjoint equations backward in time, within a gradient-based optimization loop. Thanks to the iSVD algorithm, the computational cost of solving the unsteady adjoint equations is reduced considerably, without practically affecting the accuracy of the computed gradient. Approximations to the unsteady flow fields are constructed while solving the time-varying flow equations (moving forward in time) and used to reconstruct these fields during the backward-in-time integration of the continuous adjoint equations. Optimization results obtained using the proposed method are compared to those computed using the binomial checkpointing technique, which acts as the reference method. Test cases for both flow control and shape optimization problems are presented.

Keywords: incremental SVD, adjoint optimization, unsteady continuous adjoint, approximation model, flow control, shape optimization

1 Introduction

In many applications, numerical solvers to predict time-varying phenomena (e.g. flows around bluff bodies) are used. In gradient-based optimization assisted by the adjoint method, such an unsteady flow (primal) solver is associated with the numerical solution of the unsteady adjoint PDEs, which need to be solved backward in time. During the numerical solution of the latter, the primal fields must be available at each time-step. To do so, either the re-computation of each primal field at each time-step, starting from the initial state, or the storage of all the instantaneous primal field solutions can be used. The former leads to substantial additional computational cost whereas the latter is memory-intensive.

As a remedy to this problem, various methods have been proposed to overcome the aforementioned limitations. These can be categorized as exact or sub-optimal control methods[1]. The most known representatives of the exact methods are the checkpointing techniques[2,3], being nowadays in widespread use. A fixed number of checkpoints is allocated along the time span of the primal solution and the fields of the primal variables are stored at these instants. When the adjoint PDEs are solved at a time instant at which the primal variables are not available, the primal solver starts from the nearest/previous available checkpoint, marching forward until this time instant. The number of checkpoints is decided by considering the available memory. The most efficient checkpointing variant is the dynamic binomial one[4], which ensures that, for the predefined number of checkpoints, the minimum number of re-computations is performed. It involves a binomial distribution of checkpoints in time and their dynamic (re-)allocation during the adjoint run. Even though binomial checkpointing has the minimal cost for a given memory allocation, its computational time could be proven to be substantial when compared with full storage. In[5], the solution of the adjoint PDEs takes 5 times more than that of the primal equations, though, this number may vary depending on the problem and the number of checkpoints.

Industrial unsteady optimization problems often involve small time-steps and large computational grids. In order to overcome limitations regarding memory and computational time, alternatives based on approximations to the time-varying flow field can be devised. Examples are the receding horizon control, nonlinear frequency domain and order reduction methods[6]. The idea of the receding horizon techniques is to break the original time-dependent optimization problem expressed in the elongated time horizon to a number of optimization (sub-)problems defined in shorter time intervals. In an optimization problem involving an unsteady supersonic flow[6], it was demonstrated that the computed gradients were in good agreement with those computed by an exact method.

In nonlinear frequency domain methods, also known as Harmonic Balance methods[7], the flow equations are expressed in the frequency domain, through a Fourier series transformation. Then, the solution of the flow equations takes place in the frequency domain or, through an inverse Fourier transformation, in the time domain, by converting them to a system of coupled steady-state PDEs. The adjoint to this system is, then, derived and solved. The method has been tested in the optimization of compressible viscous flows around airfoil configurations using the discrete adjoint method[8]. A Harmonic-Balance-based optimization solver was also used to increase the aero-thermal and aero-elastic performance of a transonic compressor stage rotor[9]. In addition, a frequency domain method was applied for the minimization of an helicopter rotor torque, using the initial values of the thrust level and drag force as constraints[10].

Models which aim at decreasing the degrees of freedom of the flow problem under consideration[11], known as Reduced Order Models (ROMs), can be viewed as sub-optimal control methods. The order-reduction is achieved by projecting the flow equations to a properly generated subspace, created by means of pre-computed flow snapshots. Among the available methods, Proper Orthogonal Decomposition (POD) has been widely used in fluid problems[11,12,13]. POD applies to the aforementioned set of snapshots and keeps only the predominant part of the underlying information. A POD method for generating an aerodynamic database through a parameter variation for several test cases, analyzing the number of modes required, was presented in[12]. In[13], 'gappy' data sets are considered before being used for an inverse airfoil design. POD in conjunction with Galerkin projection is used for constrained optimization problems in[14]. In[15], the POD projection subspace is built by using data from a single state and sensitivity analysis, thus alleviating the computational cost for the construction of the POD basis.

Here, a model based on the incremental variant of Singular Value Decomposition (iSVD) is proposed as a CPU and memory efficient way to approximate the instantaneous flow fields during the solution of the unsteady adjoint PDEs. The application of the SVD to the snapshots matrix containing the flow solutions of all time-steps would require the storage of the complete time-series for all flow variables at each computational node (full storage). However, this is not possible for most industrial applications. Therefore, in this work, the incremental variant of the SVD is used, alongside with an order-reduction scheme. In brief, the proposed algorithm involves the following: The left and right singular vector matrices as well as the diagonal matrix containing the singular values are updated "incrementally" at each time-step of the primal solver, based on the user-defined (reduced) rank. Thus, the projection subspace is constructed during the numerical solution of the time-varying Navier–Stokes equations and, then, used for the primal field reconstruction during the adjoint loop. This process significantly reduces the computational cost of a checkpoint-based optimization. By using the proposed model, very good approximations of the flow fields are generated and the optimization results are in good agreement with those computed by an exact method.

The paper is organized as follows; First, the derivation of the unsteady continuous adjoint method is given in brief. Then, the iSVD model is presented by focusing on its implementation within the optimization process. Finally, relevant results are discussed, demonstrating that the proposed model could indeed support unsteady adjoint-based optimization at a low computational cost. All flows considered in this work are periodic. Therefore, it is straightforward to define the objective function within a period. Nevertheless, the proposed method could also be used in conjunction with non-periodic flows, like the ones often emerging from the usage of DES or LES. There, however, the definition of the objective function window requires some investigation which was not within the scope of the paper.

2 The Unsteady Continuous Adjoint Method

This section presents in brief the development of the unsteady continuous adjoint method for periodic laminar flows of incompressible fluids. The interested reader can find more on continuous adjoint to turbulence models in[16]. In general, when an objective function is defined along the boundaries S of the domain Ω, it can be written as

$$J = \int\limits_T \int\limits_S j_S dS dt = \int\limits_T \int\limits_S j_{S,i} n_i dS dt \tag{1}$$

with n_i standing for the components of the unit normal vector pointing to S, and T for the period of the flow phenomenon. Then, its differentiation w.r.t. the vector of design variables $\boldsymbol{b}$ yields

$$\frac{\delta J}{\delta \boldsymbol{b}} = \int\limits_T \int\limits_S \frac{\delta j_{S,i}}{\delta \boldsymbol{b}} n_i dS dt + \int\limits_T \int\limits_S j_{S,i} \frac{\delta (n_i dS)}{\delta \boldsymbol{b}} dt \tag{2}$$

or by using the chain rule to develop $\frac{\delta j_{S,i}}{\delta \boldsymbol{b}}$ w.r.t. the primal variables,

$$\frac{\delta J}{\delta \boldsymbol{b}} = \int\limits_T \int\limits_S \frac{\partial j_{S,i}}{\partial v_k} n_i \frac{\delta v_k}{\delta \boldsymbol{b}} dS dt + \int\limits_T \int\limits_S \frac{\partial j_{S,i}}{\partial p} n_i \frac{\delta p}{\delta \boldsymbol{b}} dS dt + \int\limits_T \int\limits_S \frac{\partial j_{S,i}}{\partial \tau_{kj}} n_i \frac{\delta \tau_{kj}}{\delta \boldsymbol{b}} dS dt + \int\limits_T \int\limits_S j_{S,i} \frac{\delta (n_i dS)}{\delta \boldsymbol{b}} dt \tag{3}$$

where $v_i,\ i = 1, 2(, 3)$ are the velocity vector components, p is the pressure divided by the constant density and $\tau_{ij} = \nu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)$ the stress tensor components.

For periodic flows, the augmented objective function is defined as

$$L=J+\int\limits_T\int\limits_\Omega u_i R_i^v \; d\Omega dt+\int\limits_T\int\limits_\Omega qR^p \; d\Omega dt \tag{4}$$

where u_i are the adjoint velocity components, q is the adjoint pressure and R_i^v and R^p are the residuals of the unsteady Navier–Stokes equations of an incompressible fluid flow, given by

$$\begin{aligned} R^p &= -\frac{\partial v_j}{\partial x_j}=0 \\ R_i^v &= \frac{\partial v_i}{\partial t}+v_j\frac{\partial v_i}{\partial x_j}+\frac{\partial p}{\partial x_i}-\frac{\partial \tau_{ij}}{\partial x_j}=0 \end{aligned} \tag{5}$$

The differentiation of eq. (4) yields

$$\frac{\delta L}{\delta \boldsymbol{b}}=\frac{\delta J}{\delta \boldsymbol{b}}+\int\limits_T\int\limits_\Omega \left(u_i\frac{\delta R_i^v}{\delta \boldsymbol{b}}+q\frac{\delta R^p}{\delta \boldsymbol{b}}\right) d\Omega dt+\int\limits_T\int\limits_\Omega \left(u_iR_i^v+qR^p\right)\frac{\delta\left(d\Omega\right)}{\delta \boldsymbol{b}}dt \tag{6}$$

The residuals of the primal equations over the computational domain Ω are zero, thus the last term on the RHS of eq. (6) is eliminated.

Since $\frac{\delta}{\delta \boldsymbol{b}}$ includes the effect of changes in the computational domain due a change in $\boldsymbol{b}$, $\frac{\delta}{\delta \boldsymbol{b}}$ and $\frac{\partial}{\partial x_j}$ do not permute. Instead, they are linked through[17,18]

$$\frac{\delta}{\delta \boldsymbol{b}}\left(\frac{\partial (\cdot)}{\partial x_j}\right)=\frac{\partial}{\partial x_j}\left(\frac{\delta (\cdot)}{\delta \boldsymbol{b}}\right)-\frac{\partial (\cdot)}{\partial x_k}\frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta \boldsymbol{b}}\right) \tag{7}$$

After applying eq. (7) and the Gauss divergence theorem to the developed form of eq. (6), the unsteady adjoint field equations are derived by setting the field integral multipliers of $\frac{\delta v_i}{\delta \boldsymbol{b}}$ and $\frac{\delta p}{\delta \boldsymbol{b}}$ to zero. Their expressions are[16]

$$\begin{aligned} R^q&=-\frac{\partial u_i}{\partial x_i}=0 \\ R_i^u&=-\frac{\partial u_i}{\partial t}+u_j\frac{\partial v_j}{\partial x_i}-\frac{\partial\left(v_ju_i\right)}{\partial x_j}-\frac{\partial \tau_{ij}^a}{\partial x_j}+\frac{\partial q}{\partial x_i}=0 \end{aligned} \tag{8}$$

with $\tau_{ij}^a=\nu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)$ being the adjoint stress tensor components.

The adjoint boundary conditions are derived by setting the multipliers of $\frac{\delta v_i}{\delta \boldsymbol{b}}$, $\frac{\delta p}{\delta \boldsymbol{b}}$ and $\frac{\delta \tau_{ij}}{\delta \boldsymbol{b}}$ in the surface integrals of the developed form of eq. (6) to zero, where necessary. The interested reader may refer to[16] for the detailed derivation. Here, due to the time-varying nature of the flow problems, in addition to the boundary conditions, an initialization for the unsteady adjoint problem needs to be determined. The term containing the time derivative in eq. (6) can be developed as

$$\int\limits_T\int\limits_\Omega u_i\frac{\delta}{\delta \boldsymbol{b}}\left(\frac{\partial v_i}{\partial t}\right) d\Omega dt=\left[\int\limits_\Omega u_i\frac{\delta v_i}{\delta \boldsymbol{b}}\; d\Omega\right]_{t=0}^{t=T}-\int\limits_T\int\limits_\Omega \frac{\partial u_i}{\partial t}\frac{\delta v_i}{\delta \boldsymbol{b}}\; d\Omega dt \tag{9}$$

The second integral on the RHS of eq. (9) gives rise to the time derivative present in eq. (8). For $t=0$, the initial conditions for the primal velocity are imposed, resulting in $\frac{\delta v_i}{\delta \boldsymbol{b}}=0$. Since no information is known for v_i at $t=T$, the initial condition for the adjoint velocity field needs to be set as

$$u_i|_T=0 \tag{10}$$

Eq. (10) implies that the initial condition for the adjoint problem is defined at the end-time of the primal problem, thus the unsteady adjoint equations are integrated backward in time.

Having satisfied the adjoint field equations and the corresponding adjoint initial and boundary conditions, any terms remaining in eq. (6), in addition to the last term on the RHS of eq. (3), yield the sensitivity derivatives. Their expression reads[18]

$$\begin{aligned} \frac{\delta L}{\delta \boldsymbol{b}}=&\int\limits_T\int\limits_{S_W} j_{S_W,i}\frac{\delta(n_idS)}{\delta \boldsymbol{b}}+\int\limits_T\int\limits_\Omega A_{jk}\frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta \boldsymbol{b}}\right)d\Omega dt-\int\limits_T\int\limits_{S_W}\left(-u_kn_k+\frac{\partial j_{S_W,k}}{\partial \tau_{lz}}n_kn_ln_z\right)\tau_{ij}\frac{\delta(n_in_j)}{\delta \boldsymbol{b}}dSdt \\ &-\int\limits_T\int\limits_{S_W}\frac{\partial j_{S_W,k}}{\partial \tau_{lz}}n_kt_l^It_z^I\tau_{ij}\frac{\delta(t_i^It_j^I)}{\delta \boldsymbol{b}}dSdt-\int\limits_T\int\limits_{S_W}\frac{\partial j_{S_W,k}}{\partial \tau_{lz}}n_kt_l^{II}t_z^{II}\tau_{ij}\frac{\delta(t_i^{II}t_j^{II})}{\delta \boldsymbol{b}}dSdt \\ &-\int\limits_T\int\limits_{S_W}\left[\frac{\partial j_{S_W,k}}{\partial \tau_{lz}}n_k(t_l^{II}t_z^I+t_l^It_z^{II})\right]\tau_{ij}\frac{\delta(t_i^{II}t_j^I)}{\delta \boldsymbol{b}}dSdt+\int\limits_T\int\limits_{S_W^{jet}}\left(v_jn_ju_i+\tau_{ij}^an_j-qn_i+\frac{\partial J_{S_W,l}}{\partial v_i}n_l\right)\frac{\delta v_i}{\delta \boldsymbol{b}}dSdt \end{aligned} \tag{11}$$

where

$$A_{jk} = -u_i v_j \frac{\partial v_i}{\partial x_k} - u_j \frac{\partial p}{\partial x_k} - \tau^a_{ij} \frac{\partial v_i}{\partial x_k} + u_i \frac{\partial \tau_{ij}}{\partial x_k} + q \frac{\partial v_j}{\partial x_k} \tag{12}$$

$j_{S_W,i}$ is the part of $j_{S,i}$ defined along the wall boundary S_W, $\mathbf{t^I}, \mathbf{t^{II}}$ are the two tangential unit vectors along S_W forming a Frenet system with $\mathbf{n}$ and S_W^{jet} is the part of S_W where the jets are defined. In shape optimization problems, where the domain and thus the CFD grid, change whenever the shape undergoes modifications, the second term on the RHS of eq. (11) containing grid sensitivities $\frac{\delta x_k}{\delta \boldsymbol{b}}$ does not vanish[18]. This term is computed analytically since a volumetric B-Splines parameterization is used for both the geometry and the computational grid[19]. For flow control problems, the wall geometry and the computational grid are not altered, nullifying all but the last integral on the RHS of eq. (11).

Having derived the final expression for the sensitivity derivatives, eq. (11), some general remarks regarding the adjoint problem should be made. The unsteady primal set of equations constitutes a hyperbolic problem in time, for which the initial conditions are defined at $t = 0$. The unsteady adjoint equations comprise a hyperbolic problem in time, though the propagation of information occurs in the reverse direction in time. The adjoint equations, eqs. (8), include primal variables which need to be available at every time instant. As previously mentioned, possible solutions are to fully store every primal variable needed or apply the binomial checkpointing method. In the following section, an approach that utilizes a reduced-order SVD to lower memory requirements, maintaining, to a large extend, the accuracy of the optimization result, is proposed. The reduced-order SVD method is presented and used in its incremental variant.

Throughout this work, for the discretization of the Navier–Stokes equations, a second-order upwind scheme is selected for the convection term, a central scheme for the diffusion term, a second-order central differencing scheme for the pressure term and a second-order, backward Euler scheme for the time derivative. Similar numerical schemes are used for the adjoint PDEs. Regarding the update of the design variables, this is conducted using steepest descent with a constant step.

3 Incremental Singular Value Decomposition (iSVD)

Let $[\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]$ be a data set[20], with $\boldsymbol{y}_j \in \mathbb{R}^m,\ 1 \leq j \leq n$. Then, $Y = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]$ forms a $m \times n$ matrix with real values, of $\text{rank}(Y) = d \leq \min\{m, n\}$. In CFD based on cell-centered finite-volumes, m is equal to the number of grid cells and n to the number of time-steps.

Assuming that the Y matrix is available, SVD ensures[21,22] the existence of sorted real numbers, $\sigma_1 \geq \ldots \geq \sigma_d > 0$ as well as orthogonal matrices U_d and V_d, such that

$$Y = U_d \Sigma V_d^T \tag{13}$$

with $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_d) \in \mathbb{R}^{d \times d}$, $U \in \mathbb{R}^{m \times d}$ and $V \in \mathbb{R}^{n \times d}$ with $U^T U = I_U$, $I_U \in \mathbb{R}^{d \times d}$ and $VV^T = I_V$, $I_V \in \mathbb{R}^{n \times n}$, with $I_{U,V}$ being the identity matrices.

Starting from eq. (13), a reduced-order representation of the data set matrix can be obtained. Indeed, for a user-defined SVD basis of rank r, with $r \leq d$, it holds,

$$Y^r = U^r \Sigma^r V^{rT} \tag{14}$$

where superscript r denotes the first $r \leq d$ columns of the U and V matrices and singular values of Σ on the RHS in eq. (13) and $Y^r \approx Y$. Practically, the reduced-order representation of the snapshot matrix is given by $r \lll d$, with substantial advantages in terms of memory allocation. A visual representation for both the SVD and the reduced-order SVD is given in figure 1.

Typical SVD algorithms require the complete snapshot matrix to be available beforehand. This requirement makes the method inappropriate for an unsteady adjoint computation in which the snapshot matrix contains the solution of the primal variables, at each cell center of the computational grid and for all time-steps. Therefore, having the snapshot matrix available is equivalent to full storage. If the latter can be afforded, applying SVD offers no competitive advantage. However, since this is rarely the case, it is preferable to use an incremental variant of the SVD algorithm[23,24]. By doing so, the reduced-order representation of the snapshot matrix is updated immediately after a new instantaneous flow field becomes available.

Let us assume that, during the time-integration of the unsteady flow equations, a new instantaneous solution field $\boldsymbol{y} \in \mathbb{R}^{m \times 1}$ has just been computed and that the iSVD method has properly been used in the previous time-steps, resulting in a decomposition as in eq. (13). In order to incorporate the just computed vector into the already available reduced-order SVD, a 'thin' SVD update[23,24,25] is performed. For the snapshot matrix $Y^r = U^r \Sigma^r V^{rT}$, the problem is mathematically formulated as

$$[U \Sigma V^T \ \ \boldsymbol{y}] = \underbrace{[U \ \ \boldsymbol{f}]}_{\hat{U}} \underbrace{\begin{bmatrix} \Sigma & \boldsymbol{h} \\ 0 & k \end{bmatrix}}_{\hat{S}} \underbrace{\begin{bmatrix} V & 0 \\ 0 & 1 \end{bmatrix}^T}_{\hat{V}} \tag{15}$$

where $\boldsymbol{f} \in \mathbb{R}^{m\times 1}$, $\boldsymbol{h} \in \mathbb{R}^{r\times 1}$ and $k \in \mathbb{R}$. The decomposition in eq. (15) does not represent an SVD, since $\hat{S}$ is no more a diagonal matrix. Let

$$\boldsymbol{h} = U^T \boldsymbol{y} \tag{16}$$

From eq. (15) it can be written

$$\boldsymbol{y} = U\boldsymbol{h} + k\boldsymbol{f} \implies k\boldsymbol{f} = \left(I - UU^T\right)\boldsymbol{y} \tag{17}$$

By setting $\|\boldsymbol{f}\|_{L_2} = 1$, eq. (17) becomes

$$k = \| \left(I - UU^T\right)\boldsymbol{y}\|_{L_2} \tag{18}$$

Finally, combining eq. (17) and (18),

$$\boldsymbol{f} = \left(I - UU^T\right)\boldsymbol{y}/k \tag{19}$$

Having derived expressions for the computation of $\boldsymbol{f}$, $\boldsymbol{h}$ and k, in eq. (19), (16) and (18), respectively, matrix $\hat{S}$ in eq. (15) must be re-diagonalized. Therefore, by the SVD of $\hat{S}$, we obtain

$$\hat{S} = U_s \Sigma_s V_s^T \tag{20}$$

with $U_s,\ \mathrm{diag}(\Sigma_s),\ V_s \in \mathbb{R}^{(r+1)\times(r+1)}$. It follows that the updated matrix $Y' = [\ Y^r \mid y\ ]$ from eq. (15) is

$$Y' = U'\Sigma' V'^T \tag{21}$$

where

$$U' \leftarrow \hat{U} U_s \tag{22}$$
$$\Sigma' \leftarrow \Sigma_s \tag{23}$$
$$V'^T \leftarrow V_s^T \hat{V}^T \tag{24}$$

with $U' \in \mathbb{R}^{m\times(r+1)}$, $\Sigma' \in \mathbb{R}^{(r+1)}$, being diagonal, and $V' \in \mathbb{R}^{(n+1)\times(r+1)}$. This procedure increases the dimension of each matrix, which should be avoided. Therefore, in order to preserve the dimensions of $U',\ \Sigma',\ V'$, only the first r rows and columns are retained. It holds

$$U' \leftarrow \hat{U} U_{s[1:r,1:r]} \tag{25}$$
$$\Sigma' \leftarrow \Sigma_{s[1:r]} \tag{26}$$
$$V'^T \leftarrow V_{s[1:r,1:r]}^T V^T \tag{27}$$

with $U' \in \mathbb{R}^{m\times r}$, $\Sigma' \in \mathbb{R}^{r}$, and $V' \in \mathbb{R}^{(n+1)\times r}$.

Apart from maintaining the dimensions of all the matrices constant, the incremental variant of the SVD is resolved in $O\left((m+n)r^2 + m\right)$ operations. These are mainly allocated to the matrix multiplication, in eqs. (22) and (24), instead of $O(mnr)$ operations of a standard SVD[23,24].

Even though the selection of the reduced order r is important, there is no transparent way to predetermine it. A heuristic which evaluates the energy content of the reduced order to the total energy content of the snapshot matrix is used. This is expressed as

$$E(r) = \sum_{\mathrm{i}=1}^{\mathrm{r}} \sigma_{\mathrm{i}}^2 / \sum_{\mathrm{i}=1}^{\mathrm{d}} \sigma_{\mathrm{i}}^2 \tag{28}$$

An optimization cycle (with focus on the iSVD model incorporation) can be seen in algorithm 1. For the results presented in the following sections, the iSVD methodology described in this section is applied for each flow variable (i.e. pressure, velocity components and turbulence model variables) separately.

4 Active Flow Control Optimization: Flow Around A Cylinder

The flow around a circular cylinder is a well-documented flow case[26,27,28,29,30,31,32,33]; among others, parametric analyses regarding the Reynolds number, flow conditions and numerical schemes, exist. The periodic behavior of the flow around the cylinder, beyond a Reynolds number threshold, makes it an ideal test case to apply the unsteady adjoint-based optimization method. For the Re$\approx$40 case, the periodic in time von-Karman vortex street is generated, inducing periodic forces on the cylinder surface. The suppression of these forces, by means of active flow control jets placed over the surface of the body has been extensively studied in various works[34,35,36,37].

Here, the infinite velocity value is $U_\infty = 0.1$m/s, the cylinder diameter $D = 0.01$m and $\nu = 10^{-5}$, resulting in a Reynolds number of Re $= 100$. The computed Strouhal number[1] of the flow is St $= 0.169$, based on the period of the lift coefficient, which is in good agreement with the value of St ~ 0.165, proposed in the

[1] $\mathrm{St} = \frac{fD}{U_\infty}$, with f being the frequency of vortex shedding

Algorithm 1 Optimization cycle of the unsteady adjoint method using the iSVD (for a primal field $\boldsymbol{y}(\boldsymbol{x}, t)$)

```
$t \leftarrow 0$
Read $r$
while $t \leq T$ do                                                    ▷ Primal loop
    Solve for the new instantaneous primal variable field(s) $\boldsymbol{y}$
    iSVD update, eq. (15)
    if $t_i \leq r$ then
        Update $U'$, $\Sigma'$, $V'$, eqs. (22) to (24)
    else
        Update $U'$, $\Sigma'$, $V'$, eqs. (25) to (27)
    end if
    $t \leftarrow t + \Delta t$
end while
while $t \geq 0$ do                                                    ▷ Adjoint loop
    Reconstruct: $y_r \leftarrow U'\Sigma'V'^T$, computed in $O\left(r(m+1)\right)$ operations
    Solve the instantaneous adjoint equations using approximating field $y_r$
    $t \leftarrow t - \Delta t$
end while
Update design variable(s)
```

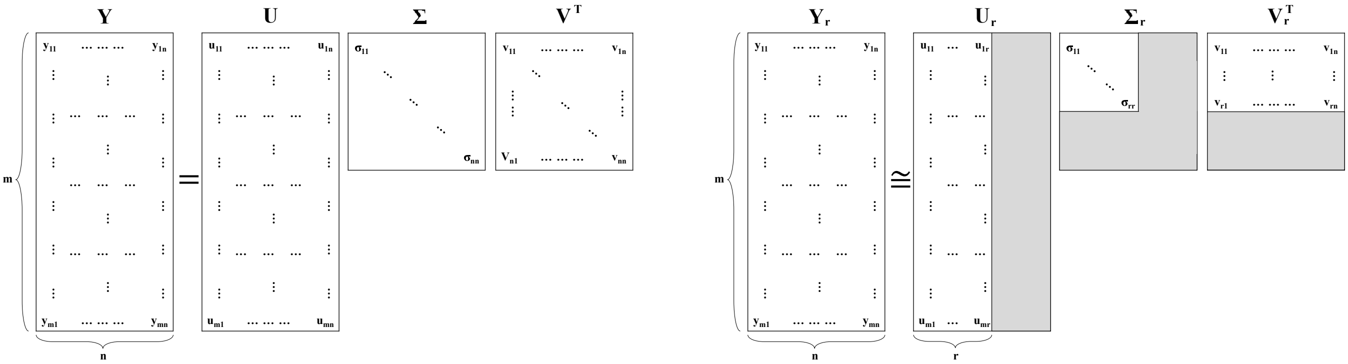


Figure 1: a) Application of the SVD on the snapshot matrix $Y \in \mathbb{R}^{m \times n}$, $n \ll m$, rank(Y)=n. b) Reconstruction of the reduced-order Y^r, $Y \approxeq Y^r$ for a user-defined rank r. Gray area stands for the omitted elements of the U, Σ and V matrices.

literature[38]. For this optimization, a 2D computational grid is used, figure 2a, with $\sim 10^4$ quadrilateral elements. The grid extends for 20D upstream and downstream and 5D up and down. A constant time-step of $\Delta t = 10^{-3}$s is considered and since the period of the phenomenon prior to the optimization is T$\sim$0.59s, $\sim$600 time-steps per period are performed. The periodic behaviour of the drag (C_d) and lift (C_l) coefficients when no flow control is applied can be seen in figure 3.

The objective function is the time-averaged squared drag force, which reads

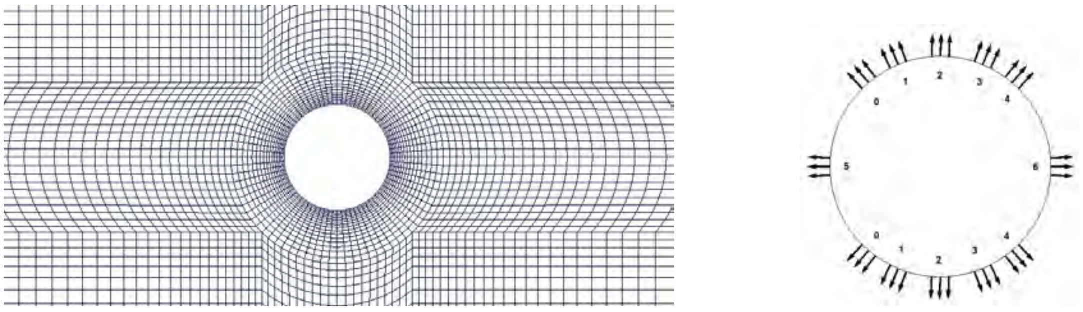
$$J_1 = \frac{1}{2T} \int_T D(t)^2 dt \tag{29}$$

$$D(t) = \int\limits_{S_W} \left(pn_i - \tau_{ij}n_j - |v_j n_j| \, v_i\right) r_i \; dS \tag{30}$$

The last term in the integrand of eq. (30) is the extra force exerted by the jet streams on the body. For the purposes of this work, pulsating jets placed along the surface of the body are used to control the flow. For each jet, a periodic velocity is employed, given by

$$v_i^m = \left(A^m \sin\left(2\pi f t - f_0\right) - A^m\right) n_i \tag{31}$$

where $m = 1, \ldots, M$ is the jet counter, A^m are the amplitudes, f the frequency and f_0 the phase of each jet. All jets are aligned with the outward, normal to the wall, unit vector n_i, $i = 1, 2$. In the present work, all jets have the same frequency and phase, thus, only the jet amplitudes A^m are considered as design variables. Positive and negative A^m correspond to blowing and suction, respectively[39,40].



(a) (b)

Figure 2: Flow around a cylinder. a) Computational grid, with $\sim 10^4$ quadrilateral elements. b) Fixed jet locations along the surface of the cylinder; five pairs of jets (0-4) and two single jets (5 and 6) are used, making 12 jets in total.

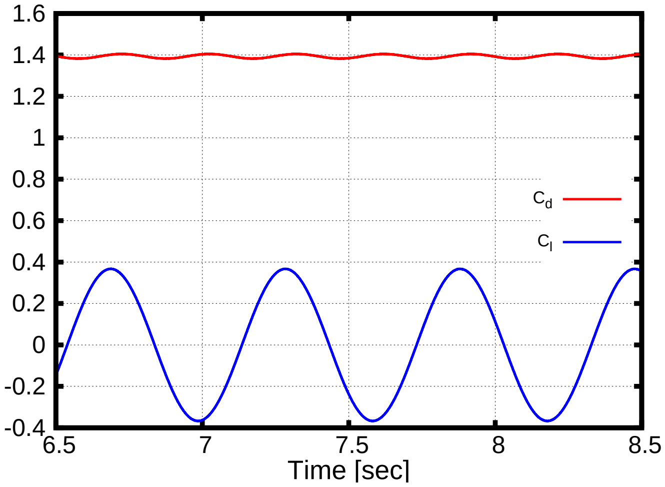


Figure 3: Flow around a cylinder. Time-series of the drag (C_d) and lift (C_l) coefficients.

Following the derivation described in section 2 the sensitivity derivatives expression reads

$$\frac{\delta L}{\delta \boldsymbol{b}} = \int_T \int_{S_W^{jet}} \left(v_j n_j u_i + \tau^a_{ij} n_j - q n_i - \frac{v_k n_k}{|v_m n_m|} v_l r_l n_i - |v_k n_k| \, r_i \right) [\sin(2\pi f t + f_0) - 1] \, n_i \, dS dt \tag{32}$$

For the flow control optimization 12 jets, are used. Their position along the cylinder circumference can be seen in figure 2b. There are five pairs of jets (IDs 0-4) with the same amplitude and two individual jets (IDs 5 and 6) which are controlled separately. The jet frequency and phase in eq. (31) are predefined as $f^m = 10$Hz and $f_0^m = 0 \ ^{\mathrm{rad}}/_{\mathrm{sec}}$. Before the optimization starts, the flow equations are solved for a sufficient time to establish periodicity and discard the transient phase. Then, each optimization cycle consists of 10 periods and starts from this flow solution. During the adjoint loop, the first 4 periods are solved to establish periodicity of the adjoint field (i.e., periods 10 to 7), while the sensitivities are computed for 4 subsequent periods (i.e., period 6 to 3). In the first optimization cycle (i.e., uncontrolled case), the flow period is $T_0 = 0.59$s; from then onward, it is assumed that the jets predominantly influence the flow, thus, the period of the flow is assumed to be equal to that assigned to the jets, i.e. $T = 0.1$s.

The convergence of the optimization is presented in figure 4a. There, it is observed that the objective function is reduced by $\sim 67\%$ after 11 optimization cycles. The C_d time-series after the optimization (i.e., termed as 'controlled'), although displaying an oscillation with a higher amplitude, has a mean value that is by $\sim 40\%$ lower than the uncontrolled case, figure 4b. Thus far, optimization results have been obtained by using the checkpointing method to reconstruct the primal fields.

Using the total energy content heuristic, eq. (28), and asking for the energy content to be $\sim 99.99\%$, only the first 15 singular values need to be kept, figure 5. Such a high energy content is a good indication that a reduced-order rank $r = 15$ is sufficient to provide a good approximation.

Let us now investigate the accuracy of the approximation that the iSVD model achieves. In figure 6a, the C_d time-series, when the iSVD model is used for the flow fields reconstruction, is shown. The reduced order model uses a rank which ranges from 5 to 15. It can be noticed that $r = 5$ results in a poor approximation, though the addition of one more singular value ($r = 6$) produces a time-series that matches the exact one in terms of amplitude but has a shift in phase, figure 6b. Further increasing the rank to $r = 15$ results in a C_d approximation which is very close to the exact time-series.

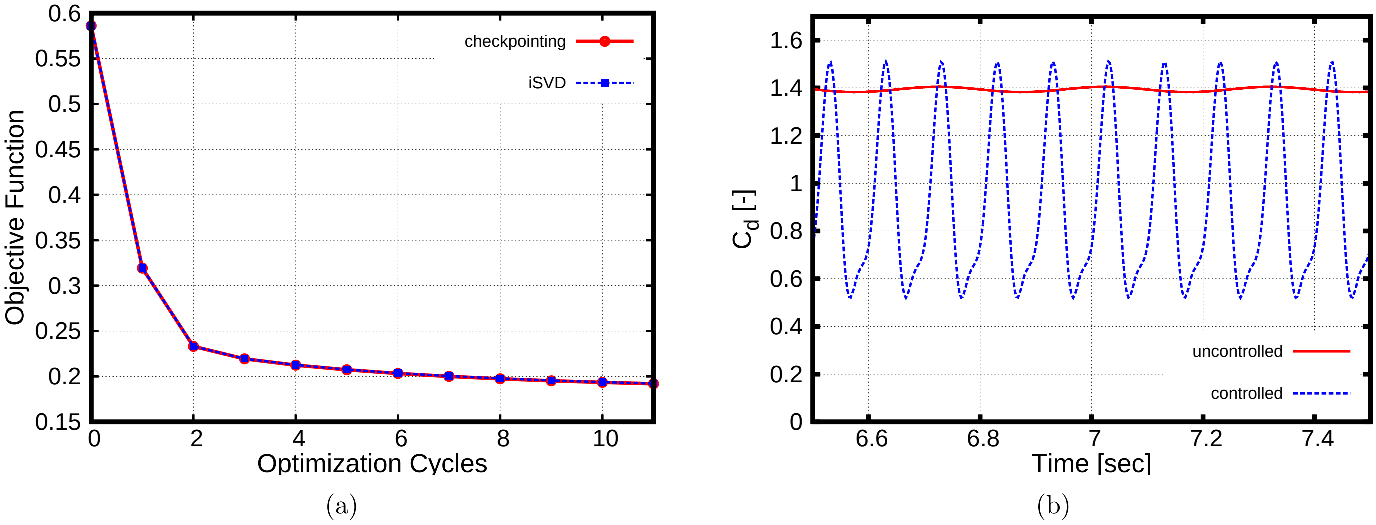


Figure 4: Flow around a cylinder. Optimization results. a) Computed objective function throughout the optimization cycles when the checkpointing and the iSVD models are used. b) C_d time-series for the uncontrolled and controlled cases.

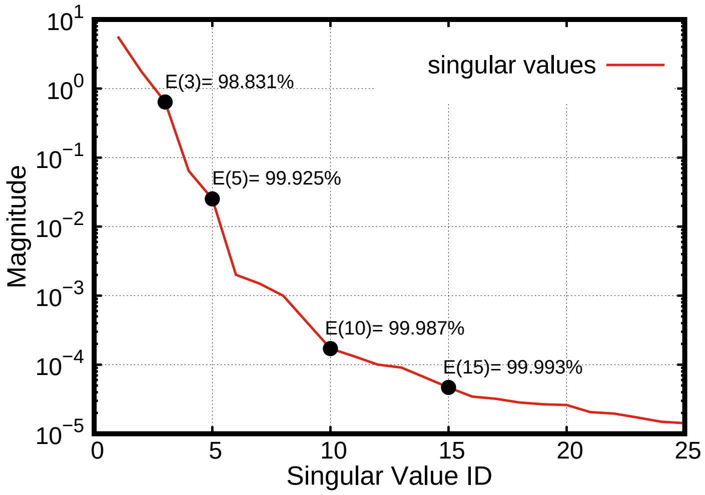


Figure 5: Flow around a cylinder. Close-up view of the singular values for the pressure full-rank snapshot matrix. The total energy content, eq. (28), is shown for a few selected ranks. The plot resulted after applying SVD to the full storage matrix.

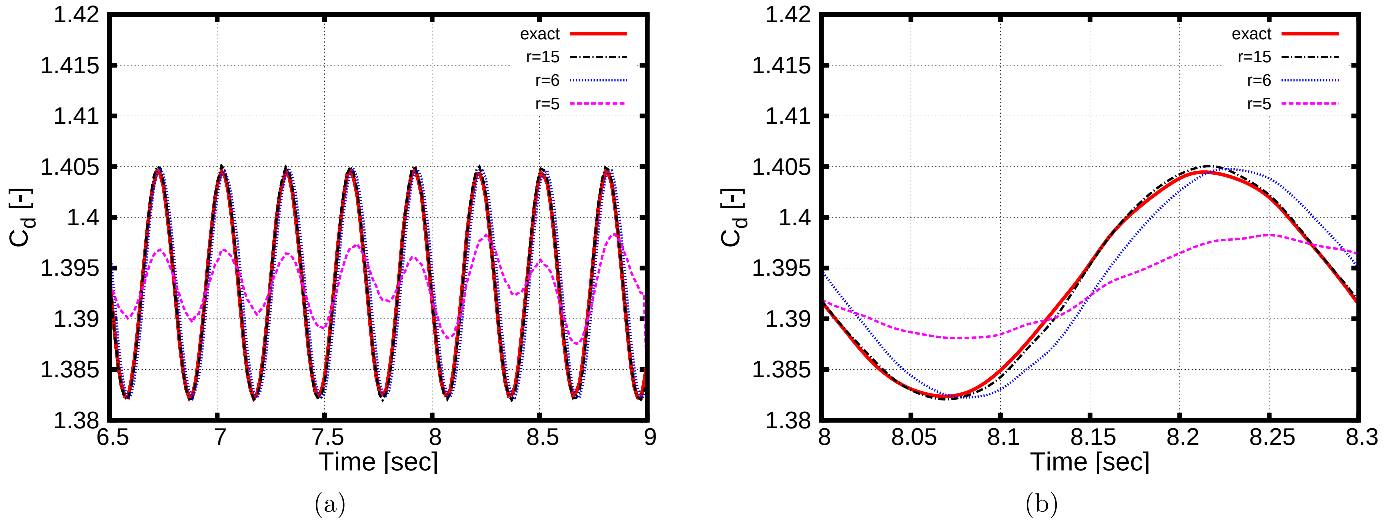


Figure 6: Flow around a cylinder. a) Approximation to the C_d time-series when the iSVD model, with $r=5$, $r=6$ and $r=15$, is used, b) Close-up view of the time-series.

Table 1: Flow around a cylinder. Optimal jet amplitudes A_m, when checkpointing and the iSVD models are used in the optimization procedure. For the iSVD model, the rank is $r = 15$.

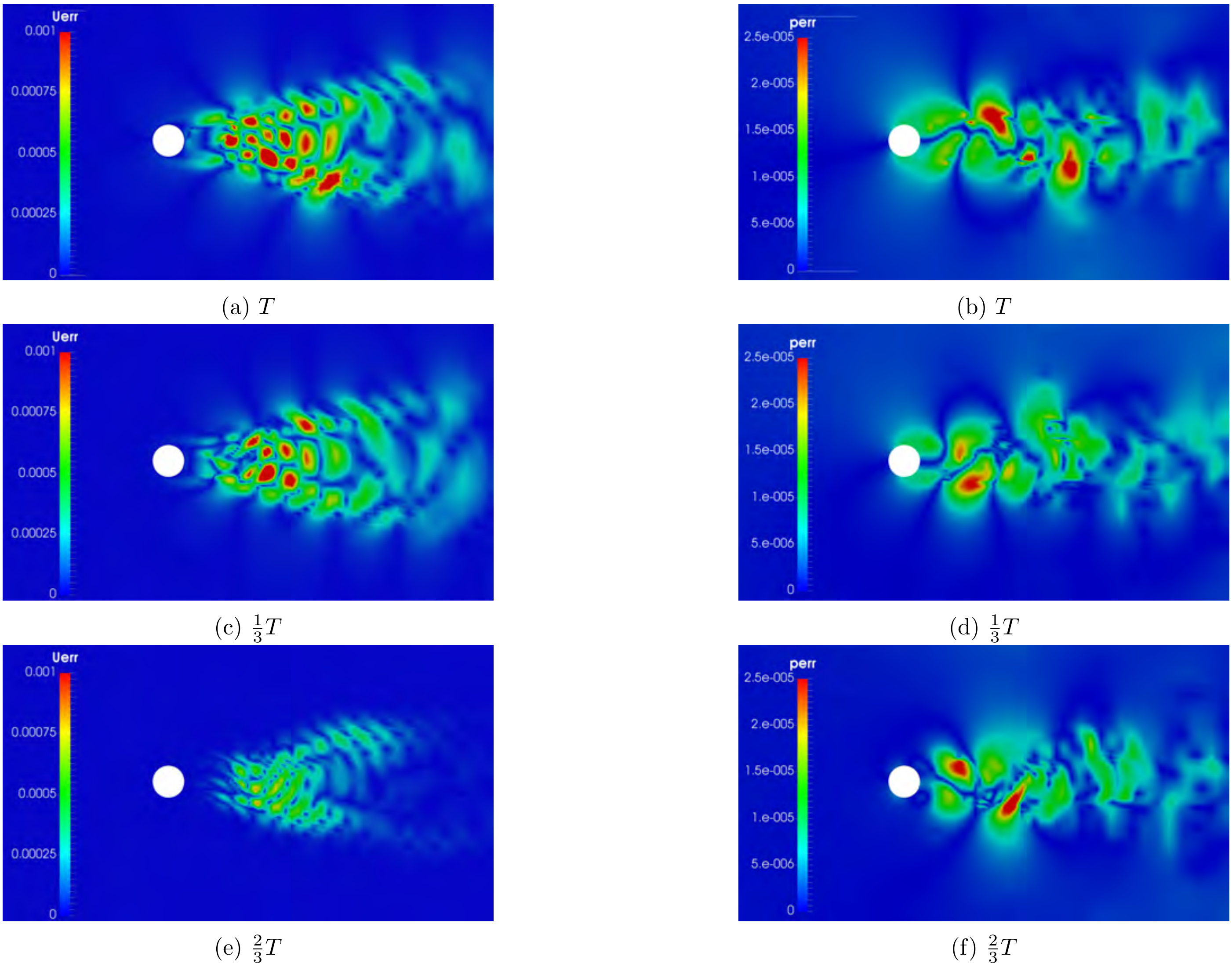
Jet	Checkpointing	iSVD	Difference
0	0.027738	0.027691	0.17%
1	-0.039552	-0.038994	1.43%
2	-0.126804	-0.126539	0.21%
3	-0.085423	-0.086517	-1.26%
4	-0.017046	-0.018128	-5.97%
5	0.082419	0.081828	0.72%
6	0.023486	0.023401	0.36%

It should be noted that, during an optimization loop, all geometries are evaluated using the time-accurate CFD solver. The comparisons depicted in figure 6 serve only as a means to evaluate the accuracy of the iSVD reconstruction. Arguably, however, for the computation of the adjoint-based sensitivity derivatives, what plays a significant role is the reconstruction error of the flow field, rather than that of the objective function. In figure 7, the error between the exact and approximated primal fields is depicted for three equidistant snapshots over one period. To quantify the error, the absolute difference between the magnitudes of the exact and approximated primal velocities is used; regarding pressure, the absolute difference between the exact and approximated primal fields is used. From figures 7a-7e, it is observed that the deviation of the approximated fields produced by the iSVD model is in the order of $\sim 0.1\%$ for the velocity (when U_∞ is used for the normalization).

A new optimization loop is then executed, based on the rank 15 iSVD (i.e., using algorithm 1), which has been demonstrated to approximate the primal fields with great accuracy. At each optimization cycle, the C_d value obtained when the iSVD model is used is practically identical to the one computed by using checkpointing, figure 4a. There, it can be noticed that for the 12^{th} optimization cycle, the iSVD model and checkpointing method result in an objective value of approximately 0.1919 and 0.1923, respectively. Therefore, the difference between the two models is approximately 0.2%. This observation is confirmed also in table 1, where it can be seen that the difference of the optimal jet amplitudes, as computed based on the iSVD and checkpointing, is in the order of $\sim 1.45\%$ on average.

Having showcased that the use of the rank 15 iSVD model leads to, practically, the same optimization results obtained by the exact approach, the next step is to compare the two approaches in terms of computational cost. For the following results, two cores of an Intel Xeon CPU E5-2630 processor at 2.40 GHz are used. The computational node used is equipped with 32 GB of memory. When the full-storage approach is considered, the first optimization cycle ($T_0 \sim 0.59$s) is completed in 728s using 11.39 GB of the total memory of the node. Using the iSVD model with $r = 15$, the first optimization cycle finishes after 811s, with excess time being primarily allocated for the subspace rotations (i.e., matrix update U' and V' in eqs. (25) and (27)) needed at each time-step of the unsteady solver. After the first optimization cycle, the period changes to $T = 0.1$s, thus, 100 time-steps are resolved in each period. Then, each optimization cycle takes ~ 153.5s and ~ 192.5s for the full-storage approach and the iSVD model, respectively. Overall, the iSVD-based optimization takes 20.9% more computational time and requires only 1.15% of the memory allocation, compared to the full-storage approach.

Although the approximation model is more expensive in terms of total computational cost when compared to the full-storage approach, the latter is hardly ever an alternative in large scale cases due to its high memory requirements. Thus, two more ‘realistic’ configurations involving checkpointing are tested. For the first one, 500 checkpoints are used. A complete optimization loop based on this approach is by 7.4% faster and 12% slower than the iSVD and full-storage alternatives, respectively. For the second configuration, 15 checkpoints are stored, which translates into the same amount of memory required by the approximation model. In this configuration, the optimization process is by 46.7% and 77.3% slower than the iSVD and the full-storage alternative, respectively.



(a) T

(b) T

(c) $\frac{1}{3}T$

(d) $\frac{1}{3}T$

(e) $\frac{2}{3}T$

(f) $\frac{2}{3}T$

Figure 7: Flow around a cylinder. Three snapshots of the approximation errors in the velocity (left) and pressure (right) fields, over one period T_0, resolved with $\sim$600 time-steps. For the approximation, the iSVD model with $r = 15$ is used.

Table 2: Flow around a cylinder. Comparison, in terms of computational cost and memory, of the first and remaining optimization cycles, for a number of approaches used to obtain the primal fields during the solution of the adjoint equations.

	Primal comp. time (s)		**Adjoint comp. time (s)**		
Approach	**1^{st} Cycle**	**Cycle¿1**	**1^{st} Cycle**	**Cycle¿1**	**Memory (GB)**
Full storage	350.00	77.50	378.00	76.00	11.39
Check/ing: 500	355.00	80.00	598.00	78.00	1.28
Check/ing: 15	350.00	77.50	1128.00	176.00	0.13
iSVD (r=15)	415.00	102.50	396.00	90.00	0.13

5 Shape Optimization of a Centrifugal Pump

This test case pertains to the European Research Community On Flow, Turbulence and Combustion (ERCOFTAC) centrifugal pump, first presented at an ERCOFTAC turbomachinery flow prediction workshop[41]. Here, a simplified 2D model, figure 8a, comprising 7 impeller blades, 12 diffuser vanes and a 6% vaneless radial gap, is used. The computational grid is block-structured, consists of $\sim 10^5$ cells, with an average value of $y^+ \sim 35$ for the first cell-centers off the wall and is illustrated in figure 8b. The rotational speed is 2000 rpm, thus, we consider a period[2] of 0.03s. The rotor tip speed is equal to $U_t = 43.98 \frac{m}{s}$, resulting in a $Re \sim 6.5 \cdot 10^6$ and the $k-\epsilon$ turbulence model is used. A sliding grid interface[42] is used for the interaction between the rotor and the stator (Rotor-Stator Interface, RSI), $S_{RSI} \equiv S_R \equiv S_S$, where, S_{RSI} is the interface, S_R and S_S the surfaces on the rotor and stator side, respectively. If the two ends of the RSI are treated as conforming boundaries with opposite normals and taking into consideration that the flow fields on these two ends are the same, developing the adjoint boundary conditions there leads to an equality of the adjoint fields across the two ends of the RSI. Hence, the interpolation scheme used in the primal solver can also be used in the adjoint one as well.

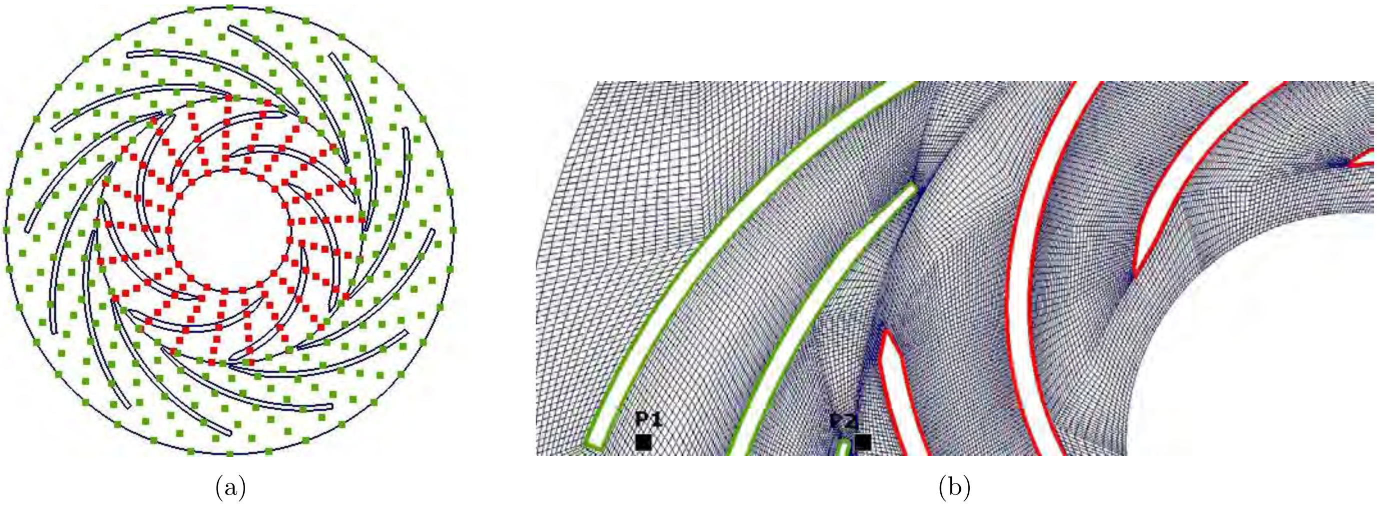


Figure 8: Centrifugal pump. a) Geometry and control points of the volumetric B-splines parameterization. The control points of the rotor and stator blades are depicted as red and green squares, respectively. b) Close-up view of the computational grid for the initial geometry. The rotor and stator blades are represented with red and green lines, respectively. The locations of the two probes, P1 (left) and P2 (right), used to monitor the pressure and velocity, are shown.

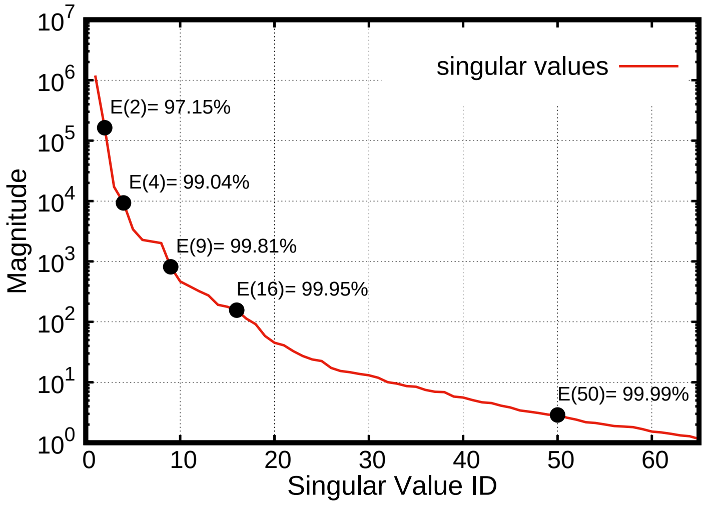
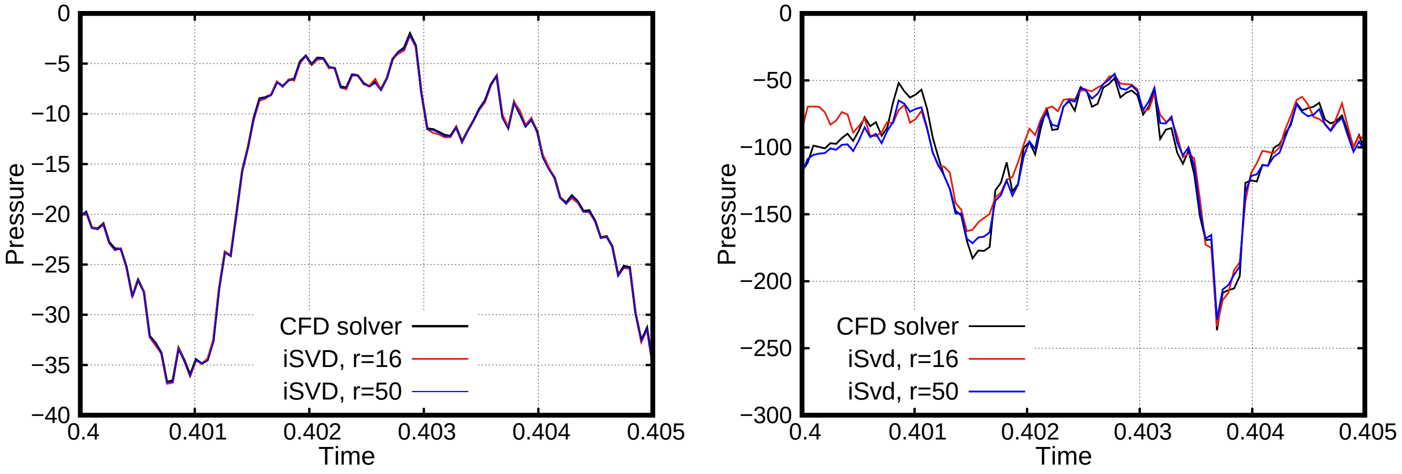


Figure 9: Centrifugal pump. Close-up view of the singular values for the pressure full-rank snapshot matrix. The total energy content, eq. (28), is shown for a few selected ranks. The plot resulted after applying SVD to the full storage matrix.

[2] The period is defined as a complete revolution of the rotor blades



(a) (b)

Figure 10: Centrifugal pump. Pressure computed by the unsteady solver (primal solution) at the two probe locations (probe 1 on the left and probe 2 on the right), figure 8b, and its approximation by the iSVD model, with r = 16 and r = 50.

The objective function to be maximized is the hydraulic efficiency; defined as

$$J_2 = 1 - \frac{H}{P} \tag{33}$$

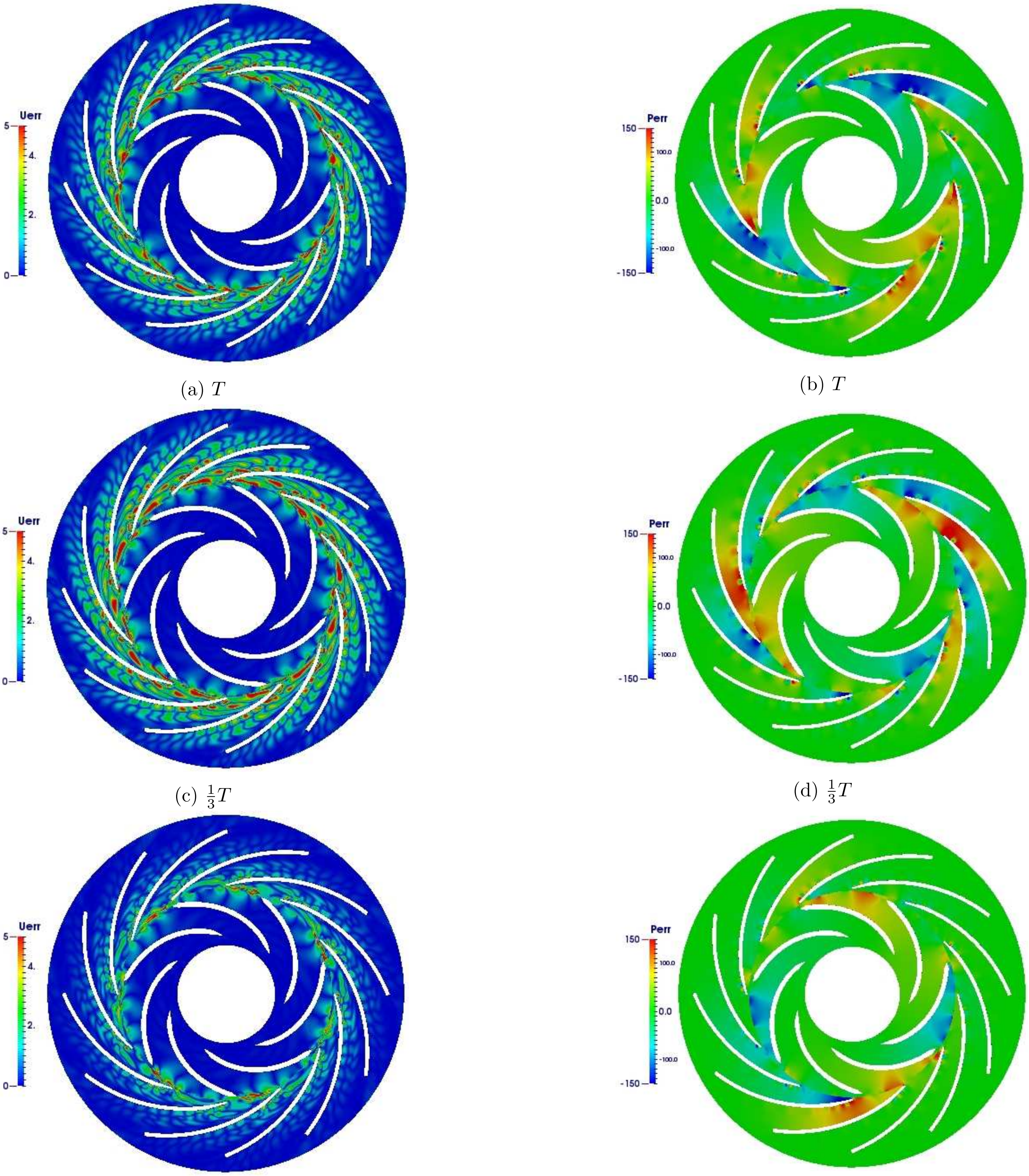
$$\begin{aligned} H &= \frac{1}{T}\int\limits_{T}\int\limits_{S_{I,O}} p_t v_k n_k \; dSdt = \frac{1}{T}\int\limits_{T}\int\limits_{S_{I,O}} H_{S^k_{I,O}} n_k \; dSdt \\ P &= \frac{1}{T}\int\limits_{T}\int\limits_{S_W} \omega_i N_i \; dSdt = \frac{1}{T}\int\limits_{T}\int\limits_{S_W} \omega_i e_{ijm} d_j \left(\tau_{mk} - p\delta_k^m\right) n_k \; dSdt \end{aligned} \tag{34}$$

where the subscripts I, O refer to the inlet and outlet of the computational domain, respectively, p_t is the total pressure, ω is the angular velocity, $\mathbf{N}$ the shaft torque, $\mathbf{d}$ a position vector, measured from an origin on the axis of rotation and e_{ijk} the Levi-Civita symbol.

Before considering the iSVD model for the retrieval of the primal fields in the adjoint solver, it is useful to use eq. (28) to get the most appropriate value for r. Figure 9 shows that the first 16 modes hold an energy content of E(16) = 99.95%, whereas E(50) = 99.99%, which only marginally improves the approximation. Indeed, examining figures 10a and 10b verifies that 16 modes provide sufficient accuracy for the primal fields retrieval. These figures correspond to the pressure reconstruction by the iSVD model, for r = 16 and r = 50, at the points seen in figure 8b. Figure 11 illustrates the error in the velocity and pressure fields for 3 equidistant time instants over one period. It can be observed that the approximated fields accurately match the primal ones in the computational grid of the rotor. In the computational domain of the stator, some discrepancies can be observed.

For the optimization, a volumetric B-splines morpher[19] is used to parameterize both the rotor and stator blades as well as the computational grid; the corresponding control points are shown in figure 8a. For the following results, 11 cores of an Intel Xeon CPU E5620 processor at 2.40 GHz are used. The primal PDEs are solved for 5 periods (periods 1 to 5); during the solution of the adjoint PDEs, the first two periods (periods 5 to 4) are used to discard the transient phase of the adjoint solution while sensitivity derivatives are computed during period 3. A constant time-step of $\Delta t = 10^{-5}$s is used, thus, 9000 primal fields are needed for the adjoint loop. Three models are considered; the checkpointing model with 3000 checkpoints and the iSVD with r = 16 and r = 50. In figure 12, it can be seen that optimizations based on all three models result in practically the same objective function value. The objective function values for the 13th cycle read as 0.92782, 0.92787 and 0.92794 for the checkpointing and iSVD with r = 50 and r = 16, respectively. Therefore, the difference of the iSVD model with r = 50 and r = 16 compared to the checkpointing exact solution is approximately 0.006% and 0.013%, respectively. Overall, the optimization increased the efficiency by ∼ 0.5%. The optimized design for the rotor blade is illustrated in figure 13a and the one for the stator blade in figure 13b.

As discussed in the previous section, the iSVD model results in a considerable saving in computational resources. This is reconfirmed in the present case as well. Using the iSVD with r = 16 leads to an overall CPU cost of 70267s for 12 optimization cycles; note that 5188s are needed for the iSVD model computations, predominantly for the multiplications in eqs. (25) and (27). When r = 50, the time needed for the same optimization is 108835s, with iSVD taking considerably longer in this setup (38610s). Running the same optimization using 3000 checkpoints (i.e. $^1/_3$ of the total amount that would be needed for full-storage),



(a) T

(b) T

(c) $\frac{1}{3}T$

(d) $\frac{1}{3}T$

(e) $\frac{2}{3}T$

(f) $\frac{2}{3}T$

Figure 11: Centrifugal pump. Three snapshots of the approximation errors in the velocity (left) and pressure (right) fields, over a single period $T = 0.03$sec. For the flow field reconstruction, the iSVD model with $r = 16$ is used.

leads to a CPU cost of 99039s. Therefore, in this test case, the iSVD model with r = 16 resulted in ∼29% lower computational cost than checkpointing, with ∼0.55% of the memory requirements.

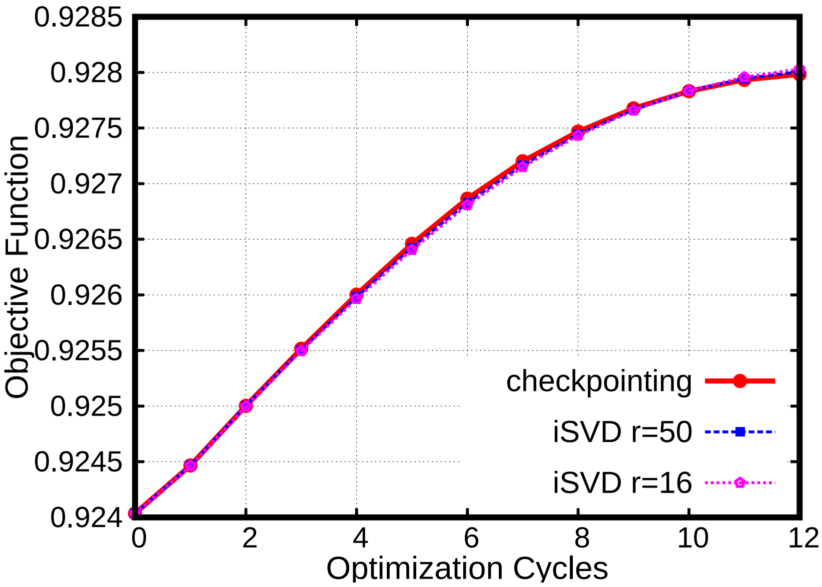
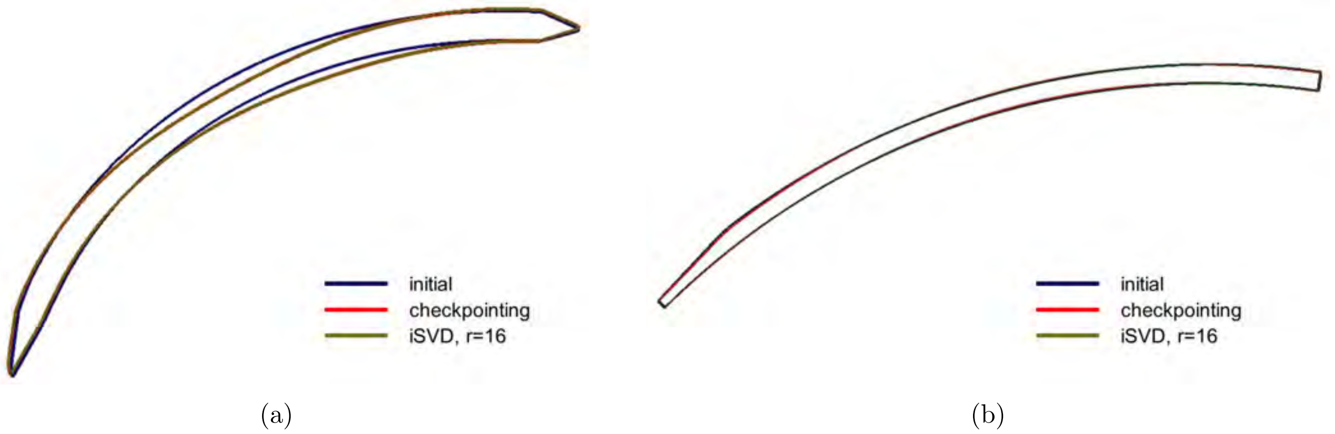


Figure 12: Convergence of the objective function during the optimization cycles; Comparison between checkpointing and the iSVD model with r = 16 and r = 50.



(a)

(b)

Figure 13: Centrifugal pump. Comparison between the initial and optimized geometries for the a) rotor and b) stator blades. It can be observed that the blades designed using the checkpointing and iSVD methods are, practically, identical.

6 Conclusions - Discussion

This paper proposes an approximation model for the primal field reconstruction, needed while integrating the unsteady adjoint equations backward in time, within an optimization loop; this model can be used as an alternative to the widely employed checkpointing method. The approximation model is based on an incremental Singular Value Decomposition of the primal solution, allowing for the on-line approximation of the flow fields while resolving the primal equations. When a new instantaneous field of the primal variables becomes available at each time-step, the incremental algorithm combines it to the (pre-)computed decomposition. Moreover, the implemented algorithm considers a reduced-order decomposition. By doing so, it results in a considerable reduction in computational memory and time. In the cylinder case, a max rank of 15 was used, allowing the flow control optimization run to be only by $\sim 2.5\%$ slower while using just $\sim 10\%$ of the memory that checkpoinitng (with 500 checkpoints) requires. When, the same memory allocation is selected for both the iSVD and checkpointing, the latter is by 51.6% slower than the former. In the centrifugal pump case, it was shown that a max rank of 16 is sufficient for the accurate shape optimization of the rotor and stator blades. In this case, the iSVD model resulted in $\sim 29\%$ less computational cost, while using $\sim 0.55\%$ of the memory that the selected checkpointing model (with 3000 checkpoints) uses. In both test cases, the optimization loop produced practically identical results when based on the proposed iSVD model and checkpointing. This tends to indicate that iSVD can efficiently support adjoint-based flow control and shape optimization processes with reduced CPU cost and memory requirements. A reliable way for a-priori choosing an appropriate iSVD rank remains an open challenge. It should be noted that the proposed method, though applied in test cases which exhibit a periodic behaviour, does not require periodicity (in contrast to Fourier series or spectral methods). Thus, the method can also be used in (purely) transient flows.

Acknowledgment

This research was funded from the People Programme (ITN Marie Curie Actions) of the European Union's 7^{th} Framework Programme (FP7/2007-2013) under REA grant agreement n^o 317006 (AboutFLOW project). The first author was an AboutFLOW Early Stage Researcher.

References

[1] Nemili A., Özkaya E., Gauger N., Thiele F., Carnarius A.. Optimal control of unsteady flows using discrete adjoints. In: 41^{st} AIAA Fluid Dynamics conference and exhibit:3720; 2011.

[2] Griewank A., Walther A.. Algorithm 799: Revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software (TOMS).* 2000;26(1):19-45.

[3] Walther A., Griewank A.. Advantages of binomial checkpointing for memory-reduced adjoint calculations. In: Numerical mathematics and advanced applications. 2004 (pp. 834–843).

[4] Wang Q., Moin P., Iaccarino G. Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation. *SIAM Journal on Scientific Computing.* 2008;31(4):2549–2567.

[5] Kapellos C., Hartmann M.. A continuous adjoint framework for vehicle aerocoustic optimization. In: 47^{th} Internoise, International Congress and Exposition on Noise Control Engineering Impact of Noise Control Engineering, Chicago, Illinois; 2018.

[6] Yamaleev N., Diskin B., Nielsen E.. Local-in-time adjoint-based method for design optimization of unsteady flows. *Journal of computational physics.* 2010;229(14):5394–5407.

[7] Giles M.. An approach for multi-stage calculations incorporating unsteadiness. In: International Gas Turbine and Aeroengine Congress and Exposition:V001T01A092 ASME; 1992.

[8] Thomas J., Hall K., Dowell E.. Discrete adjoint approach for modeling unsteady aerodynamic design sensitivities. *AIAA Journal.* 2005;43(9):1931–1936.

[9] He L., Wang D.X.. Concurrent blade aerodynamic-aero-elastic design optimization using adjoint method. *Journal of Turbomachinery.* 2011;133(1):11-21.

[10] Choi S., Lee K., Potsdam M., Alonso J.. Helicopter rotor design using a time-spectral and adjoint-based method. *Journal of Aircraft.* 2014;51(2):412–423.

[11] Walton S., Hassan O., Morgan K.. Reduced order modeling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions. *Applied Mathematical Modeling.* 2013;37(20-21):8930–8945.

[12] Mifsud M.J., Shaw S.T., McManus D.G.. A high-fidelity low-cost aerodynamic model using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids.* 2010;63(4):468–494.

[13] Bui-Thanh T., Damodaran M., Willcox K.. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal.* 2004;42(8):1505–1516.

[14] Sachs E., Volkwein S.. POD-Galerkin approximations in PDE-constrained optimization. *GAMM-Mitteilungen.* 2010;33(2):194–208.

[15] Hay A., Borggaard J., Akhtar I., Pelletier D.. Reduced-order models for parameter dependent geometries based on shape sensitivity analysis. *Journal of Computational Physics.* 2010;229(4):1327–1352.

[16] Papoutsis-Kiachagias E., Giannakoglou K.. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications. *Archives of Computational Methods in Engineering.* ;23(2):255-299.

[17] Papadimitriou D.I., Giannakoglou K.C.. A continuous adjoint method with objective function derivatives based on boundary integrals, for inviscid and viscous flows. *Computers & Fluids.* 2007;36(2):325–341.

[18] Kavvadias I., Papoutsis-Kiachagias E., Giannakoglou K.. On the proper treatment of grid sensitivities in continuous adjoint methods for shape optimization. *Journal of Computational Physics.* 2015;301:1–18.

[19] Papoutsis-Kiachagias E., Magoulas N., Mueller J., Othmer C., Giannakoglou K.. Noise reduction in car aerodynamics using a surrogate objective function and the continuous adjoint method with wall functions. *Computers & Fluids.* 2015;122:223–232.

[20] Rathinam M., Petzold L.R.. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis.* 2003;41(5):1893–1925.

[21] Noble B., Daniel J.W.. *Applied linear algebra.* Prentice-Hall New Jersey; 1988.

[22] Golub G. H., Van Loan C. F.. *Matrix computations.* Johns Hopkins University Press; 2012.

[23] Brand M.. Incremental singular value decomposition of uncertain data with missing values. *Computer Vision-ECCV 2002.* 2002;:707–720.

[24] Brand M.. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications.* 2006;415(1):20 - 30.

[25] Balzano L., Wright S.. On GROUSE and incremental SVD. In: Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5$^{\text{th}}$ International Workshop:1-4; 2013.

[26] Rahman M., Karim M., Alim A.. Numerical investigation of unsteady flow past a circular cylinder using 2D finite volume method. *Journal of Naval Architecture and Marine Engineering.* 2008;4(1):27–42.

[27] Kelkar K.M., Patankar S.V.. Numerical prediction of vortex shedding behind a square cylinder. *International Journal for Numerical Methods in Fluids.* 1992;14(3):327–341.

[28] Sohankar A., Norberg C., Davidson L.. Low-Reynolds-number flow around a square cylinder at incidence: study of blockage, onset of vortex shedding and outlet boundary condition. *International Journal for Numerical Methods in Fluids.* 1998;26(1):39–56.

[29] Sohankar A., Norberg C., Davidson L.. Simulation of three-dimensional flow around a square cylinder at moderate Reynolds numbers. *Physics of Fluids.* 1999;11(2):288-306.

[30] Gera B., Sharma P.K., Singh R.K.. CFD analysis of 2D unsteady flow around a square cylinder. *International Journal of Applied Engineering Research.* 2010;1(3):602-610.

[31] Schäfer M., Turek S., Durst F., Krause E., Rannacher R.. Benchmark computations of laminar flow around a cylinder. In: Notes on Numerical Fluid Mechanics (NNFM), vol. 48: Vieweg & Teubner Verlag 1996 (pp. 547-566).

[32] Koumoutsakos P., Leonard A.. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *Journal of Fluid Mechanics.* 1995;296:1–38.

[33] Kawamura T., Takami H., Kuwahara K.. Computation of high Reynolds number flow around a circular cylinder with surface roughness. *Fluid Dynamics Research.* 1986;1(2):145.

[34] Amitay M., Smith B. L, Glezer A.. Aerodynamic flow control using synthetic jet technology. *AIAA.* 1998;208:1998.

[35] Catalano P., Wang M., Iaccarino G., Sbalzarini I.F., Koumoutsakos P.. Optimization of cylinder flow control via zero net mass flux actuators. In: Center for Turbulence Research, Proceedings of the Summer Program. 2002 (pp. 297-303).

[36] Wang Q., Gao J.. The drag-adjoint field of a circular cylinder wake at Reynolds numbers 20, 100 and 500. *Journal of Fluid Mechanics.* 2013;730:145–161.

[37] He J.W., Chevalier M., Glowinski R., Metcalfe R., Nordlander A., Periaux J.. Drag reduction by active control for flow past cylinders. *Computational Mathematics Driven by Industrial Problems.* 2000;1739:287-363.

[38] Fey U., König M., Eckelmann H.. A new Strouhal–Reynolds-number relationship for the circular cylinder in the range $47 < Re < 2 \times 10^{-5}$. *Physics of Fluids.* 1998;10(7):1547–1549.

[39] Kavvadias I., Karpouzas G., Papoutsis-Kiachagias E., Giannakoglou K.. Optimal flow control and topology optimization using the continuous adjoint method in unsteady flows. In: In Evolutionary and Deterministic Methods For Design, Optimization and Control; 2013; EUROGEN 2013, Gran Canaria, Spain.

[40] Vezyris C., Kavvadias I., Papoutsis-Kiachagias E., Giannakoglou K.. Unsteady continuous adjoint method using POD for jet-based flow control. In: 11^{th} World Congress on Computational Mechanics, ECCOMAS. Barcelona, Spain; 2014.

[41] Combès J.. Test case U3: centrifugal pump with a vaned diffuser. In: ERCOFTAC Seminar and Workshop on Turbomachinery Flow Prediction VII, Aussois; 1999.

[42] Farrell P., Maddison J.. Conservative interpolation between volume meshes by local Galerkin projection. *Computer Methods in Applied Mechanics and Engineering.* 2011;200(1-4):89–100.