AIAA AVIATION Forum
June 15-19, 2020, VIRTUAL EVENT
AIAA AVIATION 2020 FORUM

10.2514/6.2020-3129

Computationally- and Memory-Efficient One-Shot Approach for Unsteady Adjoint-Based Design Optimization

Reza Djeddi* and Kivanc Ekici†
University of Tennessee, Knoxville, Tennessee 37996

A computationally efficient "one-shot" approach with a low memory footprint is presented for unsteady design optimization. The proposed technique is based on a novel and unique approach that combines "local-in-time" and fixed-point iteration methods to advance the unconverged primal and adjoint solutions forward and backward in time to evaluate the sensitivity of the globally time-integrated objective function. This is in some ways similar to the "piggyback" iterations where primal and adjoint solutions are evaluated at a frozen design. During each cycle, the primal, adjoint, and design update problems are solved to advance the optimization problem. This new coupled approach is shown to provide significant savings in the memory footprint while reducing the computational cost of primal and adjoint evaluations per design cycle. The method is first applied to an inverse design problem for the unsteady lid-driven cavity. Following this, vortex suppression and mean drag reduction for a circular cylinder in cross-flow is considered. Both of these objectives are achieved by optimizing the rotational speeds for steady or periodically oscillating excitations. For all cases presented in this work, the proposed technique is shown to provide significant reductions in memory as well as computational time. It is also shown that the unsteady design optimization problem converges to the same optimal solution obtained using a conventional approach.

I. Introduction

Advancements in computer science and High Performance Computing (HPC) have enabled the use of Computational Fluid Dynamics (CFD) as the standard tool for Aerodynamic Design Optimization (ADO). In the past couple of decades, the number of design parameters as well as the complexity of the CFD-based simulation tools have both increased dramatically. Such trends require the development of more robust sensitivity analysis techniques in the framework of highly sophisticated optimization tools. Even with the aid of HPC and largely accelerated CFD tools, one requires an optimization approach that can lead to the optimal design in a limited number of design cycles to reduce the computational cost. In this regard, deterministic optimization approaches, such as gradient-based search techniques, are more preferable as opposed to stochastic approaches. The reason for this choice is the fact that gradient-based techniques are capable of converging to an optimal solution in fewer design cycles, assuming that the optimality conditions are satisfied.

For gradient-based design optimization, it is necessary to have the sensitivity information evaluated in an efficient way. In many aerodynamic design applications, the number of design parameters largely exceeds the number of objective functions, which is usually a single scalar for an unconstrained design problem. Even for PDE-constrained design optimization problems, the number of objective functions, for which the sensitivity evaluation is required, would be orders of magnitude smaller than the number of design parameters. For such cases, the adjoint method would be the most efficient choice for gradient or sensitivity evaluation. In this approach, a new set of "adjoint" equations are derived by adjoining the linearized "primal" or "state"

*Research Assistant Professor and Lecturer, Department of Mechanical, Aerospace and Biomedical Engineering, Professional Member AIAA.
†Professor, Department of Mechanical, Aerospace and Biomedical Engineering, Senior Member AIAA.

equations. These equations are then solved to obtain the "co-state" solutions that can be adopted to evaluate the gradients. This specific form of the adjoint method is often referred to as the "discrete" adjoint technique. It must be noted that another form of the adjoint method, known as the "continuous" adjoint technique, has also been widely used in the community. Originally proposed by Pironneau[1] and later extended by Jameson,[2] the continuous adjoint method has some drawbacks including convergence, inconsistent errors, and numerical challenges for extending to turbulence models. The discrete adjoint approach, on the other hand, has gained a significant amount of attention in both unconstrained and PDE-constrained optimization problems in various fields of engineering design,[3–7] and is also the focus of the present work.

While adjoint methods have been widely used for aerodynamic design problems involving steady flows, their usage has been hindered in real-world applications that exhibit unsteady or even chaotic fluid flows. Optimization problems dealing with unsteady fluid responses include inverse design of excitation parameters,[8] aerodynamic shape optimization based on the minimization of a time-integrated drag coefficient,[9,10] or optimal active flow control.[11,12]

By design, the discrete adjoint method works by transposing the Jacobian of the linearized governing (or primal) equations with respect to the flow variables. This means that the adjoint-based gradient evaluation process is propagated backwards compared to the forward-propagating primal solution procedures. By the same logic, it can be easily shown that the application of the adjoint method to an unsteady flow problem would require a backward time-integration process as opposed to a forward-in-time loop used to approximate the primal solutions. Therefore, unsteady adjoint methods can become very computationally demanding while also having a huge memory footprint to store all the intermediate primal solutions.[13]

Efforts have been made to address the substantial memory overhead issues of the unsteady adjoint problems. More specifically, a "checkpointing" technique has been introduced by Griewank and Walther[14] that was based on a "divide-and-conquer" strategy proposed by Griewank.[15] In this approach, only a few snapshots of the primal solutions are stored at certain time instances (or checkpoints) during the optimization time interval. Next, during the gradient evaluation process, incremental primal solutions are evaluated and stored to back-propagate the adjoint solutions while dumping the already used primal snapshots from the memory. It has been shown that the checkpointing process can effectively reduce the memory footprint at the price of an additional computational overhead. A variation of this approach that can minimize the repetition of the dynamic checkpointing algorithm in order to reduce the added computational burden has been introduced by Wang et al.[16] More recently, Hückelheim and Müller[17] have proposed a new checkpointing approach with time gaps where interpolation techniques are used to reconstruct the primal solutions at the missing time instances. This "gappy" checkpointing approach has shown to significantly reduce the memory footprint with little to no increase in computational cost at the expense of reduced model fidelity. It must be noted that the checkpointing technique is not specific to unsteady problems but can also be used for steady design cases to reduce the memory footprint associated with the adjoint method.[14]

As described earlier, design optimization problems based on unsteady flows aim to minimize a time-integrated objective function over a desired time interval using a technique often referred to as "global-in-time." In an approach similar to checkpointing, Yamaleev et al.,[18] proposed dividing the time interval into several smaller sub-intervals where primal and adjoint calculations are performed forward- and backward-in-time, respectively. This method is referred to as "local-in-time" and has shown to significantly reduce the memory overhead since it only requires the storage of a limited number of primal solutions every time the adjoint equations are marched backward in time.[19] In another effort to reduce the storage cost of the unsteady adjoint evaluations, Beran et al.[20] proposed the use of a data reduction approach based on the Proper Orthogonal Decomposition (POD) method. In their work, the sensitivity calculations are performed based on the linearized solutions characterized by the data reduction that leads to significant savings in memory cost.

Another important aspect of the adjoint-based design optimization problems is the order with which the state, the co-state, and the design problems are solved during each design cycle. Normally, it is assumed that each design would determine a unique solution for the state or primal equations. Therefore, the objective function at each design cycle would only be a function of the design variables, thus, leading to an unconstrained optimization problem. This approach is usually referred to as *Nested Analysis and Design (NAND)* or a "reduced space" where the constraint for the existence of the state solution is treated implicitly. Thus, at the end of each design cycle, the state solution is recomputed. As a result, primal, adjoint, and design equations are solved one after the other in a "nested" fashion. The main drawback of the NAND approach is the requirement for solving the state and co-state equations using "frozen" design

variables.

As an alternative, a "full space" method has been proposed where the optimality conditions are solved for the primal, adjoint, and design variables all at the same time.[21] This monolithic approach necessitates the solution of a PDE-constrained optimization problem where the constraint for the existence of the state solution is treated explicitly. Since the primal and adjoint solutions are directly incorporated into the design optimization problem, this approach is referred to as the *Simultaneous Analysis and Design (SAND)*, which is also known as the "one-shot" method. For steady adjoint-based design optimization, the "one-shot" method can be easily implemented using an iterative fixed point solver.[22] Additionally, Ta'asan[23] and Hazra et al.[24,25] proposed a simultaneous pseudo time-stepping approach to solve the three equations (primal, adjoint, and design) derived from the optimality conditions.

In related work for transient problems, where the primal equations are solved using classical time-marching schemes, Gunther et al.[26] have proposed a modified fixed-point iteration approach that enables the use of the single-step one-shot method. While this fixed-point iteration approach can reduce the computational cost of the one-shot method for unsteady design problems, it still requires a significant amount of storage in the memory.

In the present work, a local-in-time (LiT) approach is developed to address the memory footprint issues of the unsteady adjoint-based design optimization problems. The LiT approach is utilized with varying number of sub-intervals in order to study its effects on the memory footprint and CPU times. The proposed technique is similar to the SAND approach with "piggyback" iterations where the state (primal) and the co-state (adjoint) equations are solved simultaneously by "freezing" the design variables.[22] However, the method proposed in this work requires significantly lower memory storage due to the use of smaller sub-intervals to calculate the sensitivities of a global-in-time objective. Additionally, a fixed-point iteration approach is proposed for the one-shot design optimization that greatly reduces the computational cost per design cycle. **The combination of the local-in-time and fixed-point iteration approaches – reported for the first time in the literature – leads to an extremely robust framework for time-accurate adjoint-based design optimization.** Initially, the proposed technique is applied to an inverse design problem involving the excitation parameter calibration for the lid-driven cavity flow with an unsteady lid velocity. Next, the proposed technique along with several other leading approaches in unsteady design are utilized to suppress vortex shedding and to reduce the mean drag for a circular cylinder subject to steady rotation or rotationally oscillating motion. In the following sections, details of the one-shot approach for steady and unsteady design problems as well as the governing equations for the primal CFD solver are presented followed by the optimization results.

II. One-Shot Method for Design Optimization

Although the focus of this paper is on the *unsteady design optimization*, for the sake of simplicity, the one-shot approach for the steady design optimization problem is presented first. Let us consider the minimization problem for an objective function $I(\mathbf{x}, \mathbf{Q}(\mathbf{x}))$, defined as

$$\begin{aligned} &\min\ I(\mathbf{x}, \mathbf{Q}(\mathbf{x})) \\ &\text{subject to } \mathbf{R}(\mathbf{x}, \mathbf{Q}(\mathbf{x})) = 0 \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the vector of design variables, $\mathbf{Q}$ is the vector of flow solutions, and $\mathbf{R}$ is the vector of residuals for the steady primal governing equations. Note that for an inverse design problem, the objective function can be defined simply as the difference between a quantity of interest and its target value. Using the method of Lagrange multipliers, the minimization problem in Eq. (1) can be reformulated as the Lagrangian functional such that

$$\mathcal{L}(\mathbf{x}, \mathbf{Q}, \boldsymbol{\lambda}) = I(\mathbf{x}, \mathbf{Q}(\mathbf{x})) + \boldsymbol{\lambda}^T \mathbf{R}(\mathbf{x}, \mathbf{Q}(\mathbf{x})) \tag{2}$$

where $\boldsymbol{\lambda}$ is the vector of adjoint or co-state solutions. An optimal solution for the minimization of the Lagrangian functional

$$\min\ \mathcal{L}(\mathbf{x}, \mathbf{Q}, \boldsymbol{\lambda}) \tag{3}$$

exits if the Karush-Kuhn-Tucker (KKT) optimality conditions

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Q}} = 0, \;\; \frac{\partial \mathcal{L}}{\partial \mathbf{x}} = 0, \;\; \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = 0 \tag{4}$$

are satisfied. Expanding the KKT conditions will lead to

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Q}} = \frac{\partial I}{\partial \mathbf{Q}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} = 0 \;\rightarrow\; \nabla_{\mathbf{Q}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \boldsymbol{\lambda}) = 0 \tag{5}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial I}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}} = 0 \;\rightarrow\; \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \boldsymbol{\lambda}) = 0 \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \mathbf{R}(\mathbf{x}, \mathbf{Q}(\mathbf{x})) = 0 \;\rightarrow\; \mathbf{R}(\mathbf{x}, \mathbf{Q}(\mathbf{x})) = 0 \tag{7}$$

As discussed earlier, in the one-shot approach, the system of governing equations above are solved simultaneously for the primal, adjoint, and design variables. Thus, using a Newton's method, the system of equations can be linearized as

$$\begin{bmatrix} \frac{\partial^2 I}{\partial \mathbf{Q}^2} & \frac{\partial^2 I}{\partial \mathbf{Q} \partial \mathbf{x}} & \frac{\partial \mathbf{R}}{\partial \mathbf{Q}}^T \\ \frac{\partial^2 I}{\partial \mathbf{Q} \partial \mathbf{x}} & \frac{\partial^2 I}{\partial \mathbf{x}^2} & \frac{\partial \mathbf{R}}{\partial \mathbf{x}}^T \\ \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} & \frac{\partial \mathbf{R}}{\partial \mathbf{x}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{Q} \\ \Delta \mathbf{x} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{Q}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \boldsymbol{\lambda}) \\ -\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \boldsymbol{\lambda}) \\ -\mathbf{R}(\mathbf{x}, \mathbf{Q}(\mathbf{x})) \end{bmatrix} \tag{8}$$

to determine the updates for the state, co-state, and design solutions all in the same fully-coupled framework. In order to reduce the computational cost of solving Eq. (8), Hazra et al.[25] suggested that terms $\frac{\partial^2 I}{\partial \mathbf{Q}^2}$ and $\frac{\partial^2 I}{\partial \mathbf{Q} \partial \mathbf{x}}$ can be ignored with little to no effect on accuracy. Therefore, the simplified system of one-shot governing equations can be written as

$$\begin{bmatrix} 0 & 0 & \frac{\partial \mathbf{R}}{\partial \mathbf{Q}}^T \\ 0 & \frac{\partial^2 I}{\partial \mathbf{x}^2} & \frac{\partial \mathbf{R}}{\partial \mathbf{x}}^T \\ \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} & \frac{\partial \mathbf{R}}{\partial \mathbf{x}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{Q} \\ \Delta \mathbf{x} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{Q}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \boldsymbol{\lambda}) \\ -\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \boldsymbol{\lambda}) \\ -\mathbf{R}(\mathbf{x}, \mathbf{Q}(\mathbf{x})) \end{bmatrix} \tag{9}$$

In the above equation, the Hessian of the objective function, $\frac{\partial^2 I}{\partial \mathbf{x}^2}$, used for the quadratic programming (QP) can be approximated using a few different approaches. According to Hazra et al.,[24,25] and Özkaya,[22] the Hessian can be simply replaced by: (1) a diagonal matrix for a linear steepest descent optimization; or (2) a BFGS-like approximation of the Hessian based on the gradient information, $\frac{\partial I}{\partial \mathbf{x}}$, from two subsequent design cycles for a super-linear convergence to the optimal solution. Here, another important term is the Jacobian, $\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}$, which will be discussed later in more detail when the governing equations of the primal CFD solver are presented. Next, the one-shot method for unsteady adjoint problems is discussed.

III. Unsteady Adjoint-Based Design Optimization

For unsteady design optimization problems, the global objective function is often defined as a time-integrated quantity such that

$$\overline{I} = \frac{1}{T} \int_0^T I(\mathbf{x}, \mathbf{Q}(t, \mathbf{x}))\, dt \tag{10}$$

where the transient objective function is time-averaged over an interval of $[0, T]$ with T being the final time. Similar to the steady design optimization problem, the objective function at any given time can be defined as the instantaneous difference between a quantity of interest and its target value. In contrast to the steady problem, the unsteady governing equations for the primal CFD solver will also include a time derivative (transient) term so that

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{R}(\mathbf{x}, \mathbf{Q}(t, \mathbf{x})) = 0 \quad \forall\, t \in (0, T] \tag{11}$$

Therefore, the minimization problem for the time-integrated objective function is defined as

$$\min \overline{I} = \frac{1}{T}\int_0^T I(\mathbf{x}, \mathbf{Q}(t,\mathbf{x}))\, dt \tag{12}$$
$$\text{subject to } \frac{\partial \mathbf{Q}}{\partial t} + \mathbf{R}(\mathbf{x}, \mathbf{Q}(t,\mathbf{x})) = 0 \quad \forall\, t \in (0,T]$$
$$\text{with } \mathbf{Q} = \mathbf{Q}^{\text{init}} \text{ for } t = 0$$

where the second constraint is simply the initial condition for the unsteady primal solution. The semi-discretized unsteady governing equations shown in Eq. (11) are solved using classical dual-timestepping schemes where, depending on the order of the temporal discretization, different number of primal solutions from earlier time steps are utilized. As an example, for the first-order backward difference (BDF1) and the second-order backward difference (BDF2) schemes, we have

$$\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}(\mathbf{x}, \mathbf{Q}^n(\mathbf{x})) = 0 \tag{13}$$
$$\frac{3\mathbf{Q}^n - 4\mathbf{Q}^{n-1} + \mathbf{Q}^{n-2}}{2\Delta t} + \mathbf{R}(\mathbf{x}, \mathbf{Q}^n(\mathbf{x})) = 0 \tag{14}$$

where Δt is the time-step. Assuming that the time period $[0,T]$ is divided into N equally-spaced time intervals, the time-integrated objective function, also known as the global-in-time objective, can be approximated as

$$\overline{I} = \frac{1}{T}\int_0^T I(\mathbf{x}, \mathbf{Q}(t,\mathbf{x}))\, dt \approx \frac{1}{T}\sum_{n=1}^{N} I(\mathbf{x}, \mathbf{Q}^n(\mathbf{x}))\, \Delta t \tag{15}$$

For the sake of brevity, let us only focus on the BDF1 approach and rewrite the minimization problem for the objective function (Eq. [15]) constrained by the discretized governing equations (Eq. [13]) as well as the initial condition.

$$\min \overline{I} = \frac{1}{T}\sum_{n=1}^{N} I(\mathbf{x}, \mathbf{Q}^n(\mathbf{x}))\, \Delta t \tag{16}$$
$$\text{subject to } \frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}(\mathbf{x}, \mathbf{Q}^n(\mathbf{x})) = 0 \quad \text{for } n = 2, 3, ..., N$$
$$\text{with } \mathbf{Q}^1 = \mathbf{Q}^{\text{init}}$$

Similar to the steady design optimization problem, a Lagrangian functional can be defined for the above minimization problem where

$$\mathcal{L}(\mathbf{x}, \mathbf{Q}, \boldsymbol{\lambda}) = \frac{1}{T}\left[\sum_{n=1}^{N} I(\mathbf{x}, \mathbf{Q}^n(\mathbf{x}))\, \Delta t + \sum_{n=2}^{N} [\boldsymbol{\lambda}^n]^T \left(\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n\right) \Delta t + \left[\boldsymbol{\lambda}^1\right]^T \left(\mathbf{Q}^1 - \mathbf{Q}^{\text{init}}\right)\right] \tag{17}$$

Once again, the KKT optimality conditions shown in Eq. (4) are used to derive the primal, adjoint, and design equations. In fact, considering the last optimality condition, i.e., $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = 0$, will result in the primal or "state" equations

$$\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n = 0 \quad \text{for } n = 2, 3, ..., N \tag{18}$$
$$\mathbf{Q}^1 - \mathbf{Q}^{\text{init}} = 0 \quad \rightarrow \quad \mathbf{Q}^1 = \mathbf{Q}^{\text{init}} \tag{19}$$

which are also the constraints of the unsteady minimization problem. Similarly, considering the first optimality condition, i.e., $\frac{\partial \mathcal{L}}{\partial \mathbf{Q}} = 0$, will result in the adjoint or "co-state" equations

$$\frac{\boldsymbol{\lambda}^N}{\Delta t} + \left[\frac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N}\right]^T \boldsymbol{\lambda}^N = -\frac{\partial I^N}{\partial \mathbf{Q}^N} \quad \text{terminal condition, } (n = N) \tag{20}$$

$$\frac{\boldsymbol{\lambda}^n - \boldsymbol{\lambda}^{n+1}}{\Delta t} + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T \boldsymbol{\lambda}^n = -\frac{\partial I^n}{\partial \mathbf{Q}^n} \quad \text{intermediate states, } (n = 2, ..., N-1) \tag{21}$$

$$\frac{\boldsymbol{\lambda}^1 - \boldsymbol{\lambda}^2}{\Delta t} = -\frac{\partial I^1}{\partial \mathbf{Q}^1} \quad \text{initial condition, } (n = 1) \tag{22}$$

As can be seen from Eqs. (18-19) and (20-22), the adjoint solution process involves a backward time-integration as opposed to the forward-in-time solution process of the primal system. Therefore, it is necessary to store the primal solutions, $\mathbf{Q}^n$, for the entire time period $(n = 1, ..., N)$ in the memory. These primal solutions are then used in the backward-in-time integration of the adjoint equations to determine the unsteady adjoint vectors, i.e., $\boldsymbol{\lambda}^n$, over the entire time period $(n = N, ..., 1)$, as depicted in Fig. 1. Also note that the primal and adjoint equations using the BDF2 time-marching scheme can be derived in a similar approach.

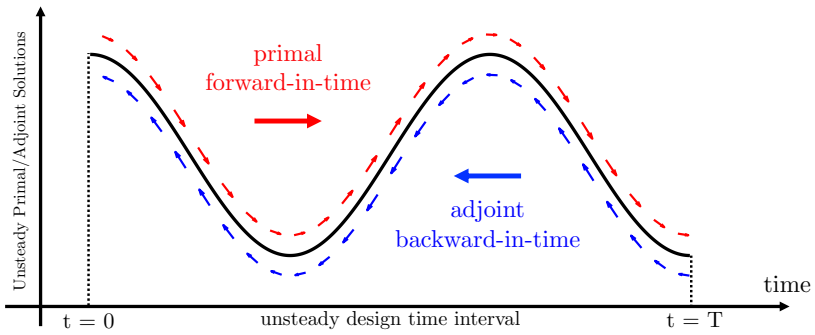


Figure 1. "Forward-in-time" procedure to obtain the primal solutions followed by the "backward-in-time" solution for the adjoints in a typical unsteady design problem over the time interval $[0,\ T]$**.**

Finally, by considering the second optimality condition, i.e., $\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = 0$, the design equation can be derived as

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{x}} =& \frac{1}{T}\sum_{n=1}^{N} \frac{\partial I^n}{\partial \mathbf{x}} \Delta t + ... \\
& \left[\frac{\partial \mathbf{Q}^N}{\partial \mathbf{x}}\right]^T \underbrace{\left(\frac{\boldsymbol{\lambda}^N}{\Delta t} + \left[\frac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N}\right]^T \boldsymbol{\lambda}^N + \frac{\partial I^N}{\partial \mathbf{Q}^N}\right)}_{=0 \quad \text{Eq. (20)}} \Delta t + ... \\
& \sum_{n=2}^{N-1} \left[\frac{\partial \mathbf{Q}^n}{\partial \mathbf{x}}\right]^T \underbrace{\left(\frac{\boldsymbol{\lambda}^n - \boldsymbol{\lambda}^{n+1}}{\Delta t} + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T \boldsymbol{\lambda}^n + \frac{\partial I^n}{\partial \mathbf{Q}^n}\right)}_{=0 \quad \text{Eq. (21)}} \Delta t + ... \\
& \left[\frac{\partial \mathbf{Q}^1}{\partial \mathbf{x}}\right]^T \underbrace{\left(\frac{\boldsymbol{\lambda}^1 - \boldsymbol{\lambda}^2}{\Delta t} + \frac{\partial I^1}{\partial \mathbf{Q}^1}\right)}_{=0 \quad \text{Eq. (22)}} \Delta t - \left[\frac{\partial \mathbf{Q}^{\text{init}}}{\partial \mathbf{x}}\right]^T \boldsymbol{\lambda}^{(1)} + ... \\
& \sum_{n=2}^{N} \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{x}}\right]^T \boldsymbol{\lambda}^n \Delta t
\end{aligned} \tag{23}$$

As can be seen in Eq. (23), the three terms inside the parenthesis vanish due to the adjoint equations. Therefore, the simplified form of the second optimality condition reads

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{1}{T}\sum_{n=1}^{N}\frac{\partial I^n}{\partial \mathbf{x}}\Delta t + \sum_{n=2}^{N}\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{x}}\right]^T \boldsymbol{\lambda}^n \Delta t - \left[\frac{\partial \mathbf{Q}^{\text{init}}}{\partial \mathbf{x}}\right]^T \boldsymbol{\lambda}^1 \tag{24}$$

which will provide the sensitivity of the Lagrangian functional with respect to the design variable. It must be noted that if the primal solutions are solved according to Eqs. (18) and (19), then the Lagrangian functional would be the same as the original time-averaged objective function, $\bar{I}$, defined in Eq. (10), i.e.,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial \bar{I}}{\partial \mathbf{x}} = \frac{1}{T}\sum_{n=1}^{N}\frac{\partial I^n}{\partial \mathbf{x}}\Delta t + \sum_{n=2}^{N}\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{x}}\right]^T \boldsymbol{\lambda}^n \Delta t - \left[\frac{\partial \mathbf{Q}^{\text{init}}}{\partial \mathbf{x}}\right]^T \boldsymbol{\lambda}^1 \tag{25}$$

A. Local-in-Time Approach for Memory Efficiency

In order to motivate the local-in-time approach developed in this work, let us consider the "piggyback" iterations technique where the following steps are taken:

1. State equations are solved forward-in-time and the primal solutions at each time step are stored in memory.
2. Co-state equations are then solved backward-in-time using the stored primal solutions.
3. Using the calculated primal and adjoint solutions, the total sensitivity of the time-integrated objective function is calculated according to Eq. (25).

Using a sequential quadratic programming (SQP) approach, the design variables can be updated by

$$\mathbf{x}^{c+1} = \mathbf{x}^{c} - \mathbf{B}_c^{-1}\left(\frac{\partial \bar{I}}{\partial \mathbf{x}}\right)_c \tag{26}$$

where c is the design cycle number and $\mathbf{B}$ is a preconditioning matrix that approximates the Hessian of the augmented Lagrangian [see Eq. (17)], or in the case of the "piggyback" iterations, the time-integrated objective given in Eq. (10), with respect to the design variables.

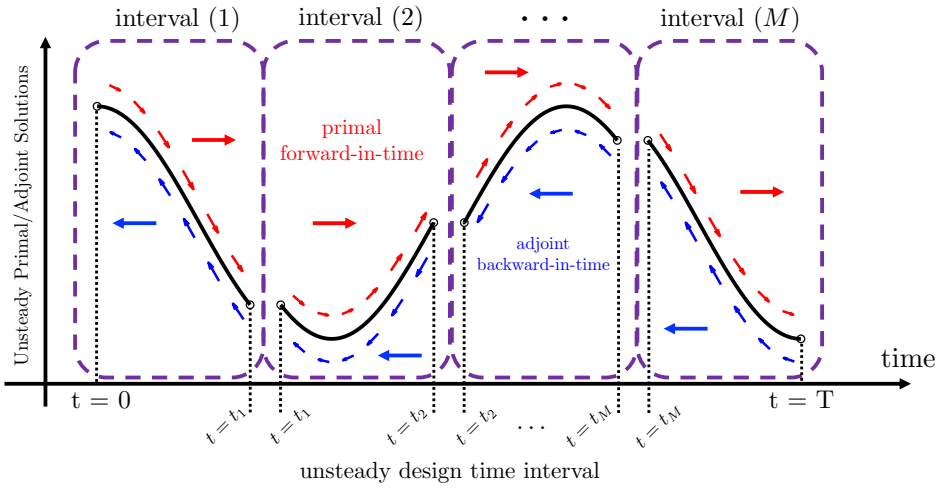


Figure 2. Design time interval $[0,\ T]$ is divided into M sub-intervals for the "Local-in-Time" approach. Here, $t_1 = \frac{T}{M}$, $t_2 = 2\frac{T}{M}$, and $t_M = (M-1)\frac{T}{M}$.

As discussed earlier, following the technique originally proposed by Yamaleev et al.,[18] the time period $[0, T]$ is divided into M smaller sub-intervals where

$$\text{interval m:} \quad (m-1)\times(\frac{T}{M}) \le t \le m\times(\frac{T}{M}) \quad \text{for } m = 1, ..., M. \tag{27}$$

Therefore, the primal equations (Eqs. [18-19]) and adjoint equations (Eqs. [20-22]) can be solved over each sub-interval instead of the entire time period as shown in Fig. 2. As can be easily seen, this approach leads to $100 \times \frac{N-(N/M)}{N}$-% reduction in memory footprint for storing primal solutions which increases if more sub-intervals are used over the time period.

B. Fixed-Point Iteration Approach for Computational Efficiency

As discussed before, the idea of fixed-point iterations for evaluating the adjoints is not limited to the unsteady design problems and have also been used for steady applications.[7] However, the idea of a one-shot approach based on the fixed-point iterations for unsteady design problems was originally proposed by Gunther et al.[26,27] In order to motivate this approach, let us once again consider the minimization of a time-averaged objective function according to Eq. (12) where the unsteady PDE-constraint is temporally discretized using a BDF1 scheme similar to Eq. (16). The constraint for this minimization problem can be rewritten in the following form

$$\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}(\mathbf{x}, \mathbf{Q}^n(\mathbf{x})) = \mathbf{R}'(\mathbf{x}, \mathbf{Q}^n(\mathbf{x}), \mathbf{Q}^{n-1}(\mathbf{x})) = 0 \quad \text{for } n = 2, 3, ..., N \tag{28}$$

where $\mathbf{R}'$ is the total residual of the unsteady primal equations. In the approach proposed by Gunther et al.,[26] the unsteady primal equations (Eq. [28]) are solved via a fixed-point iteration so that

$$\begin{aligned} &\text{loop 1: for } n = 1, ..., N \\ &\quad \text{loop 2: for } k = 1, ... \\ &\qquad \mathbf{Q}^n_{k+1} = \mathbf{F}_p(\mathbf{x}, \mathbf{Q}^n_k(\mathbf{x}), \mathbf{Q}^{n-1}(\mathbf{x})) \end{aligned} \tag{29}$$

where $\mathbf{F}_p$ is the primal fixed-point iterator designed in such a way that

$$\begin{aligned} \text{for} \quad k \to \infty \; : \quad & \mathbf{Q}^n_{k+1} = \mathbf{F}_p(\mathbf{x}, \mathbf{Q}^n_k(\mathbf{x}), \mathbf{Q}^{n-1}(\mathbf{x})) = \mathbf{Q}^n_* \\ & \text{where} \quad \mathbf{R}'(\mathbf{x}, \mathbf{Q}^n_*(\mathbf{x}), \mathbf{Q}^{n-1}(\mathbf{x})) = 0. \end{aligned} \tag{30}$$

It is worth noting that in Eq. (29), the first loop is based on the physical time steps that advances the solution forward in time while the second loop is based on the fixed-point iterations that advances the solution at each time step through pseudo-time steps or inner iterations. The idea of one-shot approach based on fixed-point iteration is to simply swap the two iteration loops such that

$$\begin{aligned} &\text{loop 1: for } k = 1, ... \\ &\quad \text{loop 2: for } n = 1, ..., N \\ &\qquad \mathbf{Q}^n_{k+1} = \mathbf{F}_p(\mathbf{x}, \mathbf{Q}^n_k(\mathbf{x}), \mathbf{Q}^{n-1}(\mathbf{x})) \end{aligned} \tag{31}$$

which means that the primal solutions are advanced in physical time even though the inner iterations have not converged yet. Gunther et al.[26,27] have mathematically proven that as long as Eq. (29) is convergent, the algorithm given in Eq. (31) will also be convergent. Therefore, the convergence of the fixed-point iteration is guaranteed.

While the idea of using a fixed-point iteration for solving the adjoint equations was proposed in the literature,[27] this work focuses on the direct solution of the adjoint equations. It must be noted that since the adjoint equations are linearized, their convergence is guaranteed for a convergent primal solution. Therefore, the solution process used in this work for the fixed-point iteration (FiP) approach is described as below with a schematic shown in Fig. 3:

1. First of all, the outer iterations (loop over k) will be treated as the loop over the design cycles. Then, for each design cycle:

2. Solve one step of the primal fixed-point iteration (Eq. [31]) for all physical time steps while storing the primal solutions in the memory.

3. Use the stored primal solutions to solve the adjoint equations (Eqs. [20-22]) backward in time.

4. Update the design variables according to the SQP formulation (Eq. [26]).

5. Stop if the gradients have been reduced below the threshold, otherwise: advance the design cycle, go back to step 2, and repeat the process.

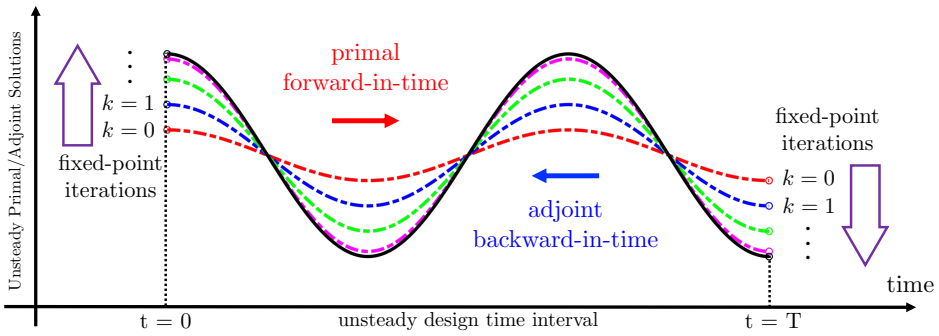


Figure 3. Progressing the "unconverged" primal solutions forward-in-time for each fixed-point iteration over the entire time interval $[0, T]$ followed by direct solutions of the linear adjoint equations in the FiP approach.

As will be shown later in the results section, the fixed-point iteration approach can greatly reduce the computational cost of the primal solver for each design cycle. On the other hand, this approach will probably increase the number of design cycles required to reach an optimal solution. However, the savings in computational cost are significant which make this approach a robust tool for unsteady adjoint-based design optimization.

C. New Hybrid Approach

Following the two approaches described earlier, this work couples the "local-in-time" gradient evaluation for memory efficiency and "fixed-point iterations" for computational efficiency. **We believe this study to be the first one reported in the literature that combines the two methods**. The idea is straightforward in the sense that each time advance loop based on the fixed-point iteration is also divided into smaller sub-intervals for forward and backward time-integration (primal and adjoint) to significantly reduce the memory footprint. The solution process for the proposed technique is depicted in Fig. 4. It must also be noted that this approach is, in some ways, similar to a checkpointing technique.

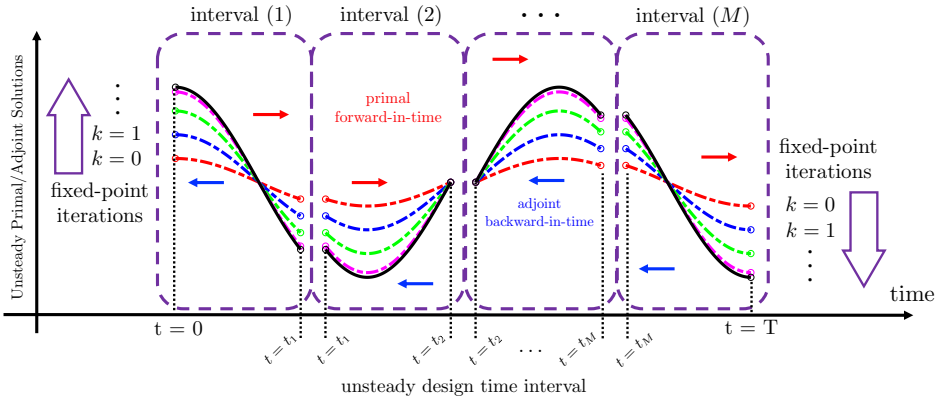


Figure 4. The proposed "hybrid" approach for unsteady design where the time interval $[0, T]$ is divided into M sub-intervals in which a fixed-point approach is utilized in order to improve both computational and memory efficiency. Here, $t_1 = \frac{T}{M}$, $t_2 = 2\frac{T}{M}$, and $t_M = (M-1)\frac{T}{M}$.

IV. Governing Equations for the Primal and Adjoint CFD Solvers

Before presenting the unsteady design optimization results, the governing equations for the primal and adjoint CFD solvers are provided in this section. As discussed earlier, two different unsteady design optimization problems are studied in this paper. The first problem, involves the flow inside a lid-driven cavity which is governed by the incompressible Navier-Stokes equations described in a rectangular domain. The second test case, however, focuses on the laminar cross-flow past a circular cylinder where incompressible Navier-Stokes equations in generalized coordinate system are considered. It must be noted that both of the flow solvers utilize the incompressible Navier-Stokes equations written based on the streamfunction and vorticity formulation. In this section, the primal and adjoint governing equations used for both of the unsteady optimization problems are presented.

A. Cavity Flow Solver

The incompressible, laminar flow inside the lid driven cavity is governed by the Navier-Stokes equations written in a vorticity-streamfunction formulation given as

$$\nabla^2\psi = -\omega \quad \text{in} \quad \mathcal{V}\times(0,T] \tag{32}$$

$$\frac{\partial\omega}{\partial t} + (\mathbf{V}\cdot\nabla)\,\omega = \frac{1}{\mathrm{Re}}\nabla^2\omega \quad \text{in} \quad \mathcal{V}\times(0,T] \tag{33}$$

where ω is the vorticity, ψ is the streamfunction, Re is the Reynolds number, and $\mathbf{V}$ is the local velocity vector [$\mathbf{V} = (u\ \vec{i} + v\ \vec{j})$ in two-dimensional Cartesian coordinates]. The physical domain $\mathcal{V}$ and the corresponding boundary and temporal conditions are presented below:

$$\begin{aligned}
\mathcal{V}(x,y,t) &: [x\in[0,L], y\in[0,L]]\times(0,T] \\
\boldsymbol{V}(x,y=L,t) &= [U_{\mathrm{lid}}(t)\ \vec{i} + 0\ \vec{j}] \quad \text{on} \quad \partial\mathcal{V}\times(0,T] \\
\boldsymbol{V}(x=0,y,t) = \boldsymbol{V}(x=L,y,t) &= \boldsymbol{V}(x,y=0,t) = 0 \quad \text{on} \quad \partial\mathcal{V}\times(0,T] \\
\boldsymbol{V}(x,y,0) &= 0 \quad \text{on} \quad \partial\mathcal{V}\times(0,T]
\end{aligned}$$

Note that the vorticity-streamfunction form of the Navier-Stokes (NS) equations only requires the solution of two equations for the two-dimensional (2D) case and it also eliminates the need for a velocity-pressure coupling, as one would normally use for an incompressible NS solvers. Based on the definition of the streamfunction, the Cartesian components of the velocity vector, $\mathbf{V}$, in 2D can be written as

$$u = \frac{\partial\psi}{\partial y} \ \text{ and } \ v = -\frac{\partial\psi}{\partial x}. \tag{34}$$

As a result, the governing equations for the primal solver can be rewritten as

$$\frac{\partial\omega}{\partial t} + \underbrace{\frac{\partial\psi}{\partial y}\frac{\partial\omega}{\partial x} - \frac{\partial\psi}{\partial x}\frac{\partial\omega}{\partial y} - \frac{1}{\mathrm{Re}}\left[\frac{\partial^2\omega}{\partial x^2} + \frac{\partial^2\omega}{\partial y^2}\right]}_{\boldsymbol{\Omega}(\psi(x,y,t),\ \omega(x,y,t))} = 0 \tag{35}$$

$$\boldsymbol{\Psi}\left(\psi(x,y,t),\ \omega(x,y,t)\right) = \omega + \left[\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2}\right] = 0 \tag{36}$$

The governing equations for the incompressible flow based on the vorticity-streamfunction formulation can be recast in the following fully-coupled form:

$$\underbrace{\frac{\partial}{\partial t}\begin{Bmatrix} 0\cdot\psi \\ \omega \end{Bmatrix}}_{\frac{\partial \boldsymbol{Q}}{\partial t}} + \underbrace{\begin{bmatrix} \frac{\partial\boldsymbol{\Psi}}{\partial\psi} & \frac{\partial\boldsymbol{\Psi}}{\partial\omega} \\ \frac{\partial\boldsymbol{\Omega}}{\partial\psi} & \frac{\partial\boldsymbol{\Omega}}{\partial\omega} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{Bmatrix} \Delta\psi \\ \Delta\omega \end{Bmatrix}}_{\boldsymbol{\Delta Q}} = \underbrace{\begin{Bmatrix} -\boldsymbol{\Psi} \\ -\boldsymbol{\Omega} \end{Bmatrix}}_{\boldsymbol{S}}, \tag{37}$$

where the Newton's method is used to solve for changes of the unknown flow variables in the inner iterations whereas a BDF2 discretization is used for advancing the solution in time. Here, $\boldsymbol{A}$ is the Jacobian matrix that can be analytically derived for a second-order linearization in space using the finite-difference method on a uniform grid.

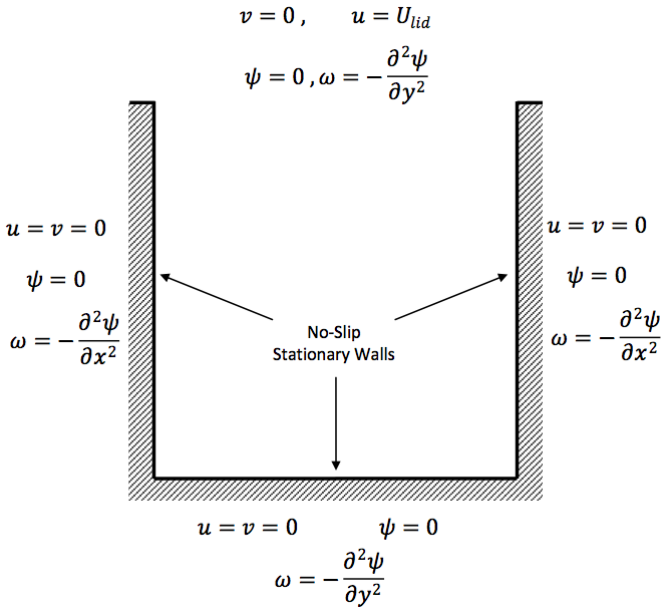


Figure 5. Boundary conditions for the primal lid-driven cavity flow solver.

Additionally, the boundary conditions for the lid-driven cavity flow problem are described in Fig. 5. As can be seen, a lid velocity is prescribed on the top boundary while no-slip stationary wall conditions are imposed on the other three boundaries. The system of equations for the primal CFD solver can be rewritten in a semi-discrete form as an unsteady PDE such that

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \boldsymbol{R}(\boldsymbol{Q}) = 0, \tag{38}$$

where $\boldsymbol{Q}$ is the vector of flow variables defined as $\boldsymbol{Q} = [\psi \;\; \omega]^T$ and $\boldsymbol{R}(\boldsymbol{Q}) = \boldsymbol{A} \cdot \boldsymbol{\Delta Q} - \boldsymbol{S}(\boldsymbol{Q})$ is the residual of the primal CFD solver. Note that Eq. (38) is the constraint PDE for the minimization problem as described earlier via Eq. (11). It is important to note that while the system of governing equations is written in the form of Eq. (38), the time-derivative should only be applied to the vorticity transport equation.

Since a BDF2 scheme is used for temporal discretization, the adjoint equations will become slightly different from those presented in Eqs. (20-22). Therefore, based on the KKT optimality condition for the "state" solution, the "co-state" equations for the adjoint vector, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_\psi \;\; \boldsymbol{\lambda}_\omega]^T$, are derived as

$$\frac{\boldsymbol{\lambda}_\omega^N}{\Delta t} + \left[\frac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N}\right]^T \boldsymbol{\lambda}_\omega^N = -\frac{\partial I^N}{\partial \mathbf{Q}^N} \quad \text{terminal condition, } (n = N) \tag{39}$$

$$\frac{\boldsymbol{\lambda}_\omega^{N-1} - \boldsymbol{\lambda}_\omega^N}{\Delta t} + \left[\frac{\partial \mathbf{R}^{N-1}}{\partial \mathbf{Q}^{N-1}}\right]^T \boldsymbol{\lambda}_\omega^{N-1} = -\frac{\partial I^{N-1}}{\partial \mathbf{Q}^{N-1}} \quad \text{semi-terminal condition, } (n = N-1) \tag{40}$$

$$\frac{\mathbf{3}\boldsymbol{\lambda}_\omega^n - 4\boldsymbol{\lambda}_\omega^{n+1} + \boldsymbol{\lambda}_\omega^{n+2}}{2\Delta t} + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T \boldsymbol{\lambda}_\omega^n = -\frac{\partial I^n}{\partial \mathbf{Q}^n} \quad \text{intermediate states, } (n = 2, ..., N-2) \tag{41}$$

$$\frac{3\boldsymbol{\lambda}_\omega^1 - 4\boldsymbol{\lambda}_\omega^2 + \boldsymbol{\lambda}_\omega^3}{2\Delta t} = -\frac{\partial I^1}{\partial \mathbf{Q}^1} \quad \text{initial condition, } (n = 1) \tag{42}$$

where $\boldsymbol{\lambda}_\omega$ is the adjoint of the vorticity field. Since the streamfunction equation (Eq. [36]) does not have an unsteady term, the adjoints of the streamfunction, $\boldsymbol{\lambda}_\omega$, can be directly evaluated via

$$\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T \boldsymbol{\lambda}_\psi^n = -\frac{\partial I^n}{\partial \mathbf{Q}^n} \qquad \text{for} \quad n = N, ..., 1 \tag{43}$$

As discussed before, the above equations are solved backward in time to obtain the adjoint solutions. Finally, the global sensitivity of the augmented Lagrangian will be evaluated using Eq. (25).

B. Incompressible Navier-Stokes Solver in Generalized Coordinate System

For the second case involving the laminar cross-flow past the circular cylinder, the incompressible Navier-Stokes equations shown earlier in Eqs. (35) and (36) are rewritten in a generalized coordinate system through a transformation from Cartesian coordinates (x, y) to computational coordinates (ξ, η) such that[28]

$$\frac{\partial \omega}{\partial t} + J\left(\frac{\partial \psi}{\partial \eta}\frac{\partial \omega}{\partial \xi} - \frac{\partial \psi}{\partial \xi}\frac{\partial \omega}{\partial \eta}\right) - \frac{1}{\mathrm{Re}}\left(\alpha\frac{\partial^2 \omega}{\partial \xi^2} + 2\gamma\frac{\partial^2 \omega}{\partial \xi \partial \eta} + \beta\frac{\partial^2 \omega}{\partial \eta^2} + P\frac{\partial \omega}{\partial \xi} + Q\frac{\partial \omega}{\partial \eta}\right) = 0 \tag{44}$$

$$\boldsymbol{\Psi}\left(\psi(x,y,t),\ \omega(x,y,t)\right) = \omega + \alpha\frac{\partial^2 \psi}{\partial \xi^2} + 2\gamma\frac{\partial^2 \psi}{\partial \xi \partial \eta} + \beta\frac{\partial^2 \psi}{\partial \eta^2} + P\frac{\partial \psi}{\partial \xi} + Q\frac{\partial \psi}{\partial \eta} = 0 \tag{45}$$

where $J = \partial(\xi,\ \eta)/\partial(x,\ y) = (x_\xi y_\eta - x_\eta y_\xi)^{-1}$ is the Jacobian of the transformation. Also, the auxiliary coefficients α, β, γ, P, and Q are described as[28]

$$\alpha = \xi_x^2 + \xi_y^2, \quad \beta = \eta_x^2 + \eta_y^2, \quad \gamma = \xi_x\eta_x + \xi_y\eta_y, \quad P = \xi_{xx} + \xi_{yy}, \quad Q = \eta_{xx} + \eta_{yy} \tag{46}$$

The incompressible Navier-Stokes equations presented in the generalized coordinate system (Eqs. [44] and [45]) can be rearranged into a fully-coupled matrix form similar to that shown previously in Eq. (37) for the cavity flow solver.

In order to solve these governing equations, boundary conditions must be imposed at the far-field and near-field boundaries. For the flow past the circular cylinder, these boundaries involve the free-stream at the far-field and the no-slip wall on the surface of the cylinder where the former dictates a zero vorticity and a constant free-stream velocity at the far-field such that (u_∞, v_∞) = (U_∞, 0.0). Assuming a unit velocity at the far-field, i.e., $U_\infty = 1.0$, and based on the definition of the streamfunction shown previously in Eq. (34), the flow variables at the far-field boundary are defined as

$$\omega_\infty = 0.0 \quad \psi_\infty = y. \tag{47}$$

At the same time, the no-slip wall condition requires a constant and zero streamfunction distribution on the surface of the cylinder, i.e., $\psi_{\text{wall}} = \dfrac{\partial \psi}{\partial \xi} = 0.0$, reducing Eq. (45) to

$$\omega(x,y) + \beta\frac{\partial^2 \psi(x,y)}{\partial \eta^2} + Q\frac{\partial \psi(x,y)}{\partial \eta} = 0 \quad \text{for } x = y = r \tag{48}$$

where r is the radius of the cylinder. The above equation can be rearranged to solve for the vorticity value on the surface of the cylinder. Additionally, for the rotating cylinder cases considered in this work, the angular surface velocity can be written in generalized coordinate system as

$$u_\theta = -\frac{\partial \psi}{\partial r} = -\left[\frac{\partial \psi}{\partial \xi}\frac{\partial \xi}{\partial r} + \frac{\partial \psi}{\partial \eta}\frac{\partial \eta}{\partial r}\right] \tag{49}$$

The first term on the right-hand-side of the above equation must vanish due to the no-slip wall condition on the surface of the cylinder thus, simplifying it as

$$u_\theta = -\left[\frac{\partial \psi}{\partial \eta}\left(\frac{\partial \eta}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial \eta}{\partial y}\frac{\partial y}{\partial r}\right)\right] \quad \rightarrow \quad \frac{\partial \psi}{\partial \eta} = -\frac{u_\theta}{\eta_x\cos(\theta) + \eta_y\sin(\theta)} \tag{50}$$

where "Polar" to "Cartesian" coordinate transformations, i.e., $x = r\cos(\theta)$ and $y = r\sin(\theta)$, are utilized to replace the $\frac{\partial x}{\partial r}$ and $\frac{\partial y}{\partial r}$ terms. Ultimately, the rotational velocity on the surface of the cylinder, $u_{\text{rotational}} = u_\theta$, can be prescribed as

$$u_{\text{rotational}} = \Omega \qquad \text{for the "steady" rotation case}$$
$$u_{\text{rotational}} = \Omega \, \sin(2\pi \, f \, t) \qquad \text{for the "periodic" rotation case}$$

where Ω is the amplitude of rotation and f is the non-dimensional forcing frequency which is analogous to the Strouhal (St) number defined as

$$\text{St} = \frac{2r \cdot f}{U_\infty} \tag{51}$$

for a cylinder with a unit diameter, i.e., $r = 0.5$, subject to a unit free-stream velocity, $U_\infty = 1.0$.

In this work, an in-house CFD solver named GENESIS is used to discretize and solve the incompressible Navier-Stokes equations presented here in the generalized coordinate system along with the near-field and far-field boundary conditions. The fully coupled system of equations are linearized in time using a BDF2 scheme while second-order finite differences are utilized for spatial discretization on an O-typed grid.

After obtaining the streamfunction and vorticity solutions, the static pressure field is calculated following the approach presented by Garmann.[28] Having determined the static pressures on the wall boundary, the lift and drag force coefficients can be calculated according to

$$C_L(t) = C_L^p + C_L^v = \oint \frac{2}{J} \eta_y \, p(x,y,t) d\xi + \oint \frac{1}{J} \left[-\frac{2}{\text{Re}} \, \eta_x \, \omega(x,y,t) \right] d\xi \tag{52}$$

$$C_D(t) = C_D^p + C_D^v = \oint \frac{2}{J} \eta_x \, p(x,y,t) d\xi + \oint \frac{1}{J} \left[\frac{2}{\text{Re}} \, \eta_y \, \omega(x,y,t) \right] d\xi \tag{53}$$

where superscripts p and v refer to the pressure and viscous force components, respectively, and once again the Reynolds number is defined as $\text{Re} = \frac{2rU_\infty}{\nu_\infty}$ based on the free-stream velocity, a reference length, and the kinematic viscosity (ν) of the fluid.

V. Results

In this section, the proposed one-shot approach is used for unsteady design optimization. The first test case involves the inverse design in the framework of a lid-driven cavity flow solver. Next, vortex suppression and mean drag reductions are sought for a cylinder in cross-flow subject to "steady" and "oscillatory" rotations.

A. Inverse Design in a Lid-Driven Cavity

As the first test case to demonstrate the performance of various unsteady primal/adjoint solution techniques, the determination of the excitation parameter for the unsteady lid velocity is targeted. For the lid driven cavity flow, the unsteady lid velocity is described as

$$U_{\text{lid}}(t) = \frac{1}{\sin^{-1}(K)} \sin^{-1}\left(K \sin(ft)\right) \tag{54}$$

where K is the excitation parameter and f is the excitation frequency taken to be $f = 2\pi$ in this work. The excitation parameter is usually defined between $0 \leq K \leq 1$. For smaller values of the excitation parameter, i.e., $K \to 0$, the lid velocity will have a sinusoidal form while for $K \to 1$, a triangular waveform with sharp crests is achieved.

The primal solution of the CFD solver for a single period, $T = \frac{2\pi}{f} = 1.0$, is obtained for the lid-driven cavity flow at a Reynolds number of $\text{Re} = 100$ on a rectangular grid with 101×101 equally-spaced nodes and $\Delta t = 0.05$. The time history of the vorticity solutions at the mid-point on the lid boundary, i.e., at $x = L/2$ and $y = L$, is presented in Fig. 6 for an excitation parameter of $K = 0.9999$. As can be seen, for this value of the excitation parameter, the unsteady vorticity solutions clearly exhibit a triangular waveform with sharp crests. Additionally, the primal solutions at $t = 0.25$, $t = 0.5$, $t = 0.75$, and $t = 1.0$ are presented in terms of the vorticity fields and are shown in Fig. 7.

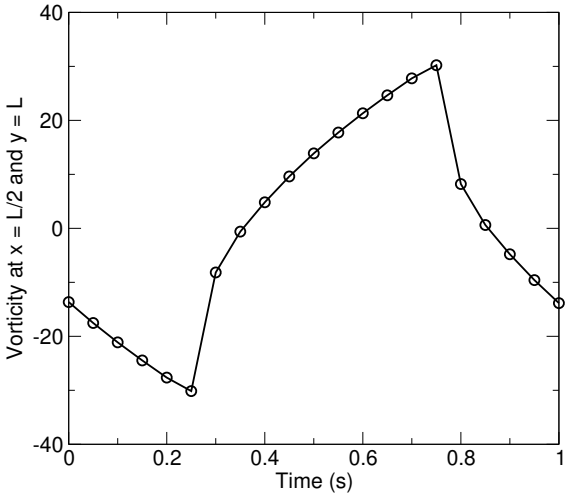
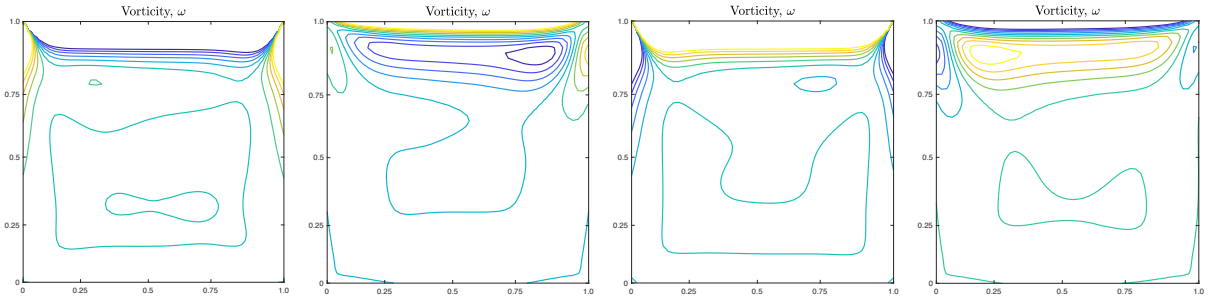


Figure 6. Time history of the vorticity response at the mid-point of the lid boundary for an unsteady lid velocity with $K = 0.9999$.



(a) $t = 0.25$ (b) $t = 0.5$ (c) $t = 0.75$ (d) $t = 1.0$

Figure 7. Primal solutions of the vorticity field at four time steps during one period of excitation, $T = \frac{2\pi}{f} = 1.0$.

For the purpose of the unsteady design optimization, the inverse design problem for the excitation parameter is considered. The time-integrated objective function is defined as

$$\overline{I} = \frac{1}{T}\int_{t=0}^{t=T} \frac{1}{2}\left(\omega(t) - \omega_{\text{target}}(t)\right)\, dt = \frac{1}{T}\sum_{n=1}^{N} \frac{1}{2}\left(\omega^n - \omega^n_{\text{target}}\right)\Delta t \tag{55}$$

where the vorticity, ω, and its target value, ω_{target}, are both defined at the mid-point of the lid boundary. The target time history of the unsteady vorticity response is taken to be the one obtained using $K = 0.9999$ as shown previously in Fig. 6. Also, the initial value of the excitation parameter is taken to be $K = 0.8$ which corresponds to a semi-sinusoidal transient response.

In the present study, the unsteady adjoint-based design optimization problem is solved using four different approaches:

1. The classical approach (**Classical**): State solutions are obtained by solving primal equations forward in time for the entire time period and the primal solutions are stored in the memory. Next, co-state solutions are obtained by solving adjoint equations backward in time for the entire time period. Finally, the global sensitivity of the time-integrated objective function is calculated and used for updating the design variables via SQP (see Fig. 1).

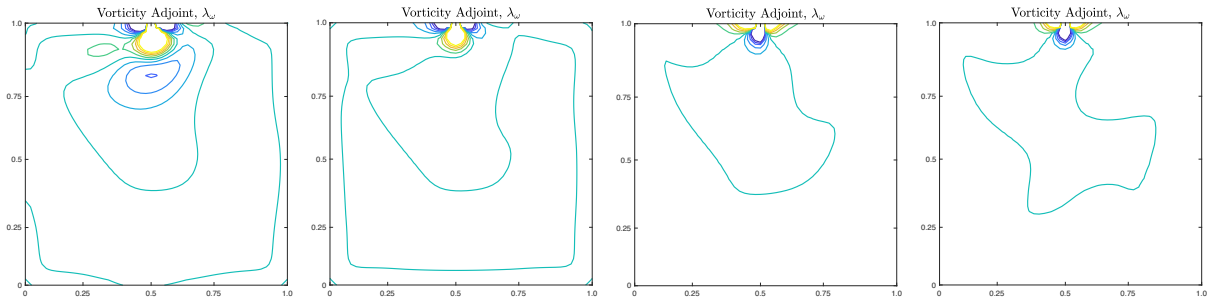
2. Local-in-Time approach (**LiT**): Similar to the classical (global-in-time) approach with the only difference that forward and backward time-marching for the primal and adjoint solutions are performed on

several smaller sub-intervals over the entire period to minimize memory footprint (see Fig. 2).

3. Fixed-Point Iteration approach (**FiP**): In each design cycle, the state solutions are obtained by solving only a single fixed-point iteration of the primal equations. The linear system of adjoint equations, on the other hand, are solved directly. In this approach, the primal solutions are advanced forward in time without explicitly driving the fixed-point iterations to convergence (see Fig. 3).

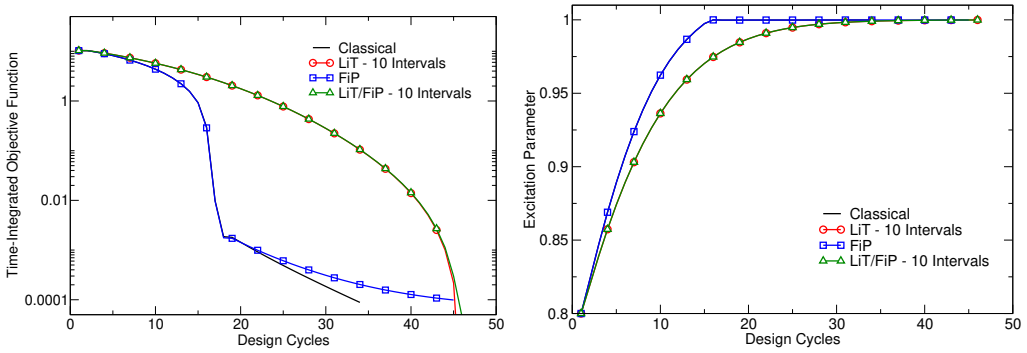
4. Proposed technique (**LiT/FiP**): In this work, we propose a unique one-shot approach for unsteady adjoint design problem. In this method, the primal equations are solved using a fixed-point iteration approach where time marching scheme advances the primal solutions forward in time. Simultaneously, the primal and adjoint equations are solved for various number of sub-intervals over the entire time period in forward and backward modes, respectively (see Fig. 4).

Before presenting the results from the one-shot design optimization approach, let us look at the adjoint solutions at four time instances during the interval of interest. These time steps are taken to be at $t = 0.25$, $t = 0.5$, $t = 0.75$, and $t = 1.0$ which correspond to the primal solutions presented earlier in Fig. 7. The adjoint solutions at these time steps are presented in Fig. 8. Since the objective function is a function of the vorticity solution at the mid-point on the lid boundary, the adjoint solution field is also focused around this point. Additionally, the flow reversal phenomenon, which is typically seen in adjoint solution, causes the sensitivities to back-propagate into the computational domain from the source of the objective function.



(a) $t = 0.25$ (b) $t = 0.5$ (c) $t = 0.75$ (d) $t = 1.0$

Figure 8. Adjoint solutions of the vorticity field at four time steps during one period of excitation, $T = \frac{2\pi}{f} = 1.0$**.**

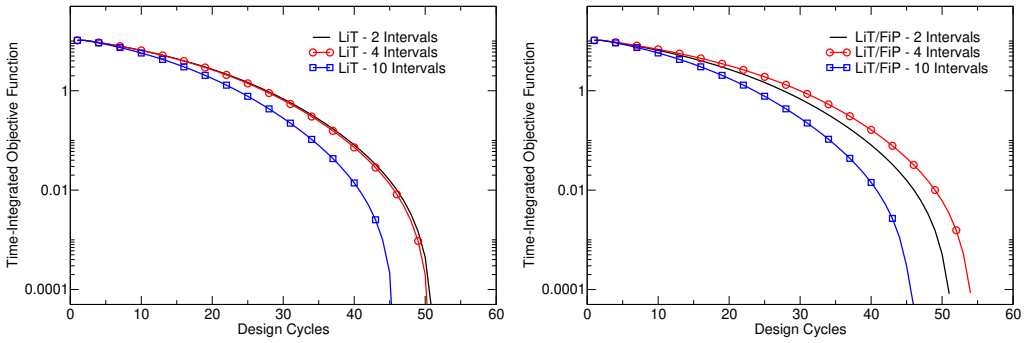


(a) Convergence of $\overline{I}$ (b) Convergence of K

Figure 9. Convergence of the time-integrated objective function (left) and the excitation parameter (right) for the unsteady design optimization problem using four different approaches.

In all cases studied here, the design cycles are continued until the objective function has reached a 0.0001 threshold. The convergence of the time-integrated objective function using the four different approaches

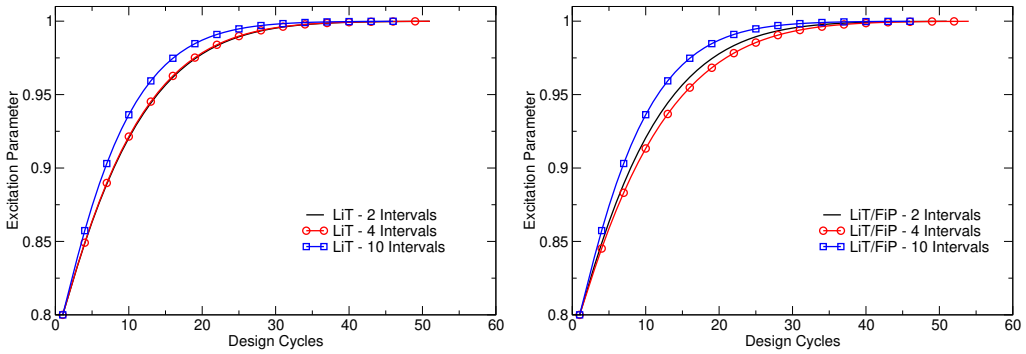
described earlier as well as the convergence of the excitation parameter (K) are presented in Fig. 9. As can be seen, the convergence of the fixed-point iteration (FiP) approach is very similar to that of the classical one-shot method with a slight difference during the last portion of the convergence. It must be noted that the current design optimization problem is bound-constrained so that infeasible values of the excitation parameter, i.e., $K > 1.0$ are not admissible. In fact, the upper bound for the design variable is activated after 18 design cycles at which point the convergence behavior of the minimization problem changes and it is no longer super-linear. On the other hand, the two cases with local-in-time approach both have a slower convergence as they slowly try to increase the excitation parameter from its initial value to the target. Overall, this results in a slower convergence to the optimal value while the convergence behavior does not change.



(a) **LiT** approach (b) **LiT/FiP** approach

Figure 10. Effects of the number of sub-intervals on the convergence of the time-integrated objective function for the unsteady design optimization problem using the LiT (left) and LiT/FiP approaches.

It must be noted that for the **LiT** and **LiT/FiP** approaches, cases with two, four, and ten sub-intervals are considered while only the results with ten sub-intervals were presented in Fig. 9. In order to study the effects of the number of sub-intervals on the performance of the unsteady one-shot design optimization approach, cases with different numbers of sub-intervals are compared against each other. These results are presented in Figs. 10 and 11 in terms of the objective function and excitation parameter convergence for the **LiT** and **LiT/FiP** approaches where local-in-time technique is utilized. Interestingly, for both the **LiT** and **LiT/FiP** approaches, cases with 10 sub-intervals are leading to a faster convergence while also providing almost 90% reduction in memory footprint.



(a) **LiT** approach (b) **LiT/FiP** approach

Figure 11. Effects of the number of sub-intervals on the convergence of the excitation parameter for the unsteady design optimization problem using the LiT (left) and LiT/FiP approaches.

Finally, the time history of the vorticity at the mid-point of the lid boundary for the original ($K = 0.8$),

optimized, and target ($K = 0.9999$) are compared against each other and the results are presented in Fig. 12. As can be seen, the inverse design optimization problem was able to perfectly match the target vorticity distribution by calibrating the excitation parameter. We remind the reader that all four approaches ultimately converge to the same target value of $K = 0.9999$ and hence, only the optimal solution from the LiT/FiP approach is presented in Fig. 12.

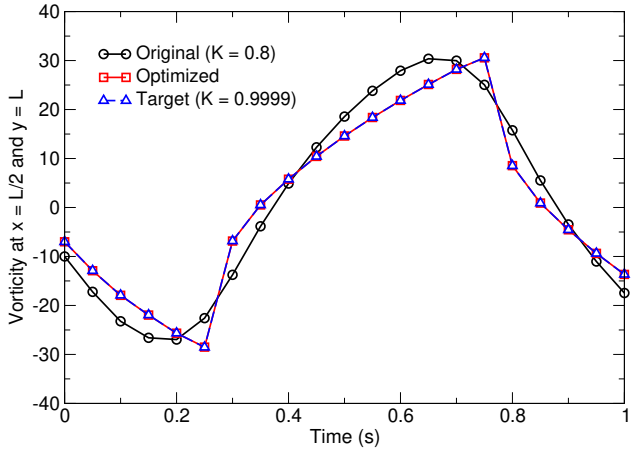


Figure 12. Comparison of unsteady vorticity solutions for the original ($K = 0.8$), optimized, and target ($K = 0.9999$) designs of the cavity flow with unsteady lid velocity.

More importantly, the fixed-point iteration approach (FiP) provides more than 40% reductions in CPU time for the evaluation of primal and adjoint solutions for the entire time interval. Combining this computational efficiency with the memory efficiency of the local-in-time (LiT) approach has shown to provide significant reductions in both CPU time and memory footprint. Therefore, the LiT/FiP approach presented in this work has the potential of providing a robust and highly efficient framework for unsteady design optimization.

B. Vortex Suppression and Mean Drag Reduction for Cylinder in Cross-Flow

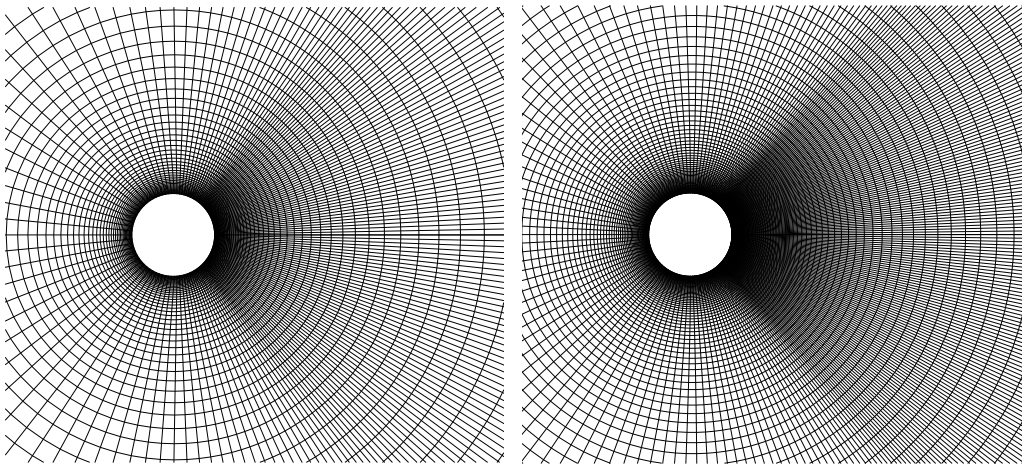
The flow past a circular cylinder has been studied extensively in the literature due to the fact that it provides essential understanding of the vortex dynamics in the wake of general bluff bodies. The characteristics of the wake flow are directly related to the Reynolds number with rows of vortices forming inside the wake as the Reynolds number is increased beyond a critical value of $\mathrm{Re}_{\mathrm{crit}} > 47$, in a so-called Karman vortex street.[29]

With a deeper understanding of the flow features in the wake of the cylinder, focus has been shifted toward controlling the vortex shedding by means of rotating the cylinder. It has been understood that rotating spheres and cylinders can lead to the suppression of the Karman vortex shedding due to the Magnus effects[30,31] which can be controlled via the angular velocity of the rotating obstacle. The experimental work of Tokumaru and Dimotakis[32] proved that finding an optimal rotational velocity for the cylinder in cross-flow can lead to significant mean drag reductions. This effect has been also studied numerically by He et al.[33,34] for various Reynolds numbers. Additionally, in the area of unsteady adjoint-based design, several studies have been focused on vortex suppression or mean drag minimization for a circular cylinder via constant or time periodic rotation of the cylinder.[12,35,36]

In this paper, the proposed unsteady one-shot approach is utilized for the same purpose to find the optimal rotational velocity that can either suppress the vortex shedding behind the cylinder or simply reduce the mean drag coefficient. Here, flow past the circular cylinder with a unit diameter at $\mathrm{Re} = 100$ is considered. For the case with "steady" rotation, the computational domain is consisted of an O-grid with 177×81 nodes that is extended for 100 diameters away from the cylinder as shown in Fig. 13(a).

1. Vortex Suppression via Steady Rotation

In the first optimal control problem for the cylinder in the cross-flow, the steady rotation is considered. Here, the rotational velocity on the surface of the cylinder is defined as



(a) 177×81 grid (b) 265×129 grid

Figure 13. Computational grids used for the flow past the circular cylinder with "steady" (on left with 177×81 nodes) and "periodic" (on right with 265×129 nodes) rotation settings. Notice the node clustering in the wake of the cylinder for both grid resolutions.

$$u_{\text{rotational}} = \Omega$$

where the rotational velocity amplitude, Ω, is chosen as the design variable for the optimal control problem. With the goal of vortex suppression via steady rotation of the cylinder, we follow an approach similar to that proposed by Homescu et al.[35] where a target flow-field is considered in the framework of a "flow-tracking" procedure. For this reason, the flow field at $\text{Re} = 2$ with a steadily rotating cylinder at $\Omega = 2$ is chosen.[35] Therefore, the objective function for the "flow-tracking" approach can be written as

$$\overline{I} = \frac{1}{T}\int_0^T \frac{1}{2}\int_{\mathcal{V}} \left[\psi - \psi^{\text{target}}\right]^2 \, d\mathcal{V}\, dt \tag{56}$$

where $\mathcal{V}$ refers to the entire computational domain and ψ^{target} is the streamfunction field for the target solution, i.e., $\text{Re} = 2$ and $\Omega = 2$. The present choice of the objective function is significantly more straightforward when compared to the mean drag coefficient. However, as shown by Homescu et al.,[35] it can lead to an ill-posed optimization problem where the optimal rotational speed would be continually increased. Therefore, a regularization function, similar to that proposed by Homescu et al.,[35] will be utilized to resolve the ill-posedness issues. The regularization function tries to minimize the rotational velocity on the surface of the cylinder by augmenting the objective function such that

$$\overline{I} = \frac{1}{T}\int_0^T \frac{1}{2}\int_{\mathcal{V}} \left[\psi - \psi^{\text{target}}\right]^2 \, d\mathcal{V}\, dt + \mathbf{R_F}\frac{1}{T}\int_0^T \frac{1}{2}\oint_{\mathcal{S}} \left[u_{\text{rotational}}\right]^2 \, d\mathcal{S}\, dt \tag{57}$$

where $\mathcal{S}$ refers to the cylinder surface boundary and $\mathbf{R_F}$ is a constant factor that controls the regularization function. It must be noted that the addition of the regularization function is in many ways similar to a "penalty function" approach. Homescu et al.[35] have shown that a trial and error technique can be used to determine the regularization factor for various Reynolds numbers. In this work, a similar approach has been pursued which has led to $\mathbf{R_F} = 1500$ for the $\text{Re} = 100$ case presented in this section.

First, the primal flow field with $\Omega = 1.0$ is solved for 150 seconds with a time-step of $\Delta t = 0.02$ to achieve quasi-steady-state conditions. Next, the unsteady design optimization problem is "hot started" and a 20 second long time interval is chosen as the design window. Similar to the unsteady inverse design problem of the lid driven cavity that was presented earlier, the classical (global-in-time), local-in-time (LiT), fixed-point iteration (FiP) approaches as well as the proposed hybrid technique known as LiT/FiP are utilized. For the

local-in-time approach varying number of sub-intervals are utilized. Here, for the design window of [0, 20], the number of sub-intervals is taken to be $M = 2$, 4, 10, and 25 for both the LiT and LiT/FiP approaches.

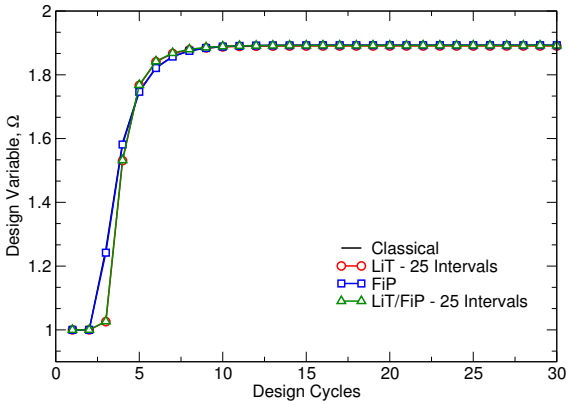
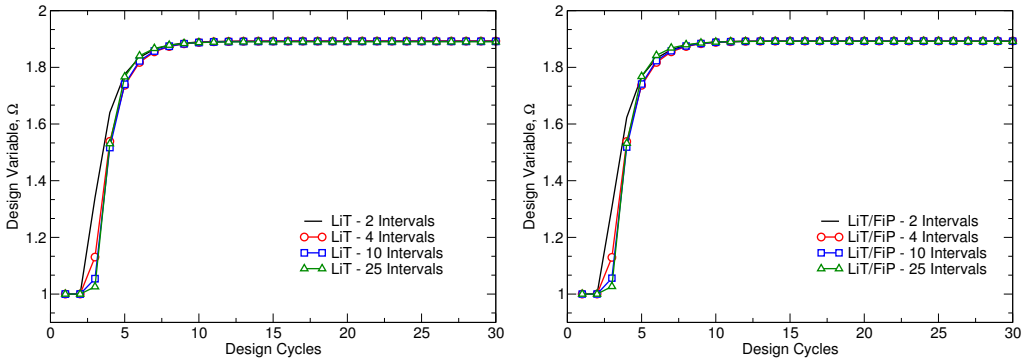


Figure 14. Convergence of the quadratic programming problem for the optimal steady rotation rate of the cylinder in cross-flow at $\mathbf{Re} = 100$.

With the choice of $\mathbf{R_F} = 1500$ for this case, the sequential least-squares quadratic programming (SLSQP) method converges to an optimal rotational velocity amplitude of $\Omega_{\text{opt}} = 1.8932$. The convergence of the optimizer using the classical approach (GiT), LiT method with 25 intervals, fixed-point iteration (FiP) approach, and the proposed LiT/FiP technique also with 25 sub-intervals are presented in Fig. 14.

As can be seen, all four approaches converge to the same optimal solution while the cases utilizing a local-in-time approach exhibit a slightly slower convergence during the early design cycles. This behavior can be associated with the fact that the LiT approach is prone to slight errors in early stages due to the discontinuities present in the backward-in-time procedure used for solving adjoint equations as the terminal conditions for these equations would rely on adjoint solutions from the next sub-interval that are yet to be solved for.



(a) **LiT** approach (b) **LiT/FiP** approach

Figure 15. Effects of the number of sub-intervals on the convergence of the rotational velocity amplitude, Ω, for the unsteady design optimization problem (cylinder in cross-flow) using the LiT (left) and LiT/FiP approaches.

Next, the effect of the number of sub-intervals on the convergence behavior of the unsteady one-shot techniques is studied. Here, the LiT and LiT/FiP approaches with various number of sub-intervals are considered and the results are presented in Fig. 15. As can be seen, while all techniques converge to the same optimal solution, the increase in the number of sub-intervals can have a slight degradation of the convergence behavior at early stages.

In order to further study the performance of different unsteady one-shot techniques, the optimality condition is plotted during the optimization process and the results are shown in Fig. 16. Here, it is interesting

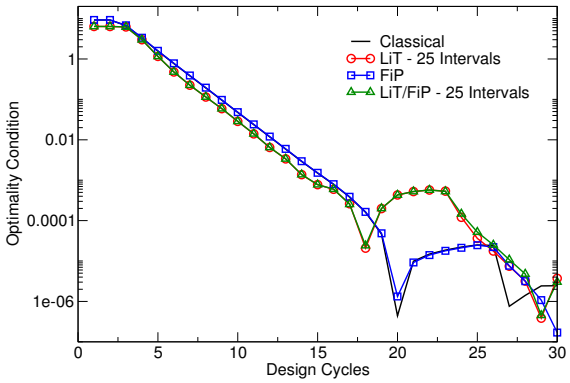


Figure 16. Optimality condition for the quadratic programming used in the determining the optimal steady rotation rate of the cylinder in cross-flow at Re = 100.

to note that in all cases, the gradients of the time-averaged objective function are reduced significantly. Additionally, the LiT and LiT/FiP techniques which rely on a local-in-time approach, exhibit a different behavior at the late stages due to the possible under-shooting and over-shooting of the optimizer close to the optimal solution. In fact, Rumpfkeil and Zingg[12] have shown that the design space for this control problem has various local optima that are tightly close to each other which can lead to such behavior.

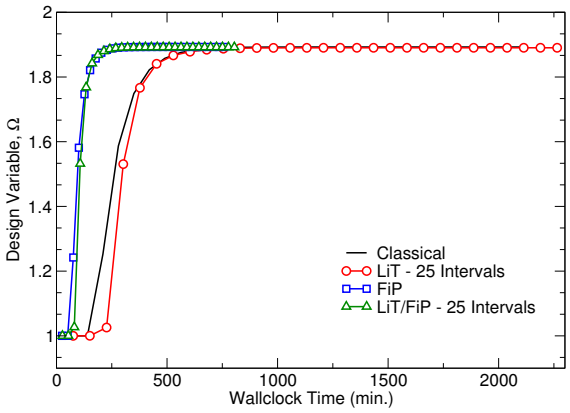


Figure 17. CPU times for different unsteady one-shot techniques used in this work for the optimal control of the vortex shedding of a circular cylinder at Re = 100 via a steady rotation.

Table 1. Normalized CPU times for different unsteady one-shot approaches used for the optimal control of the cylinder vortex shedding via steady rotation.

One-Shot Approach	Normalized Time	One-Shot Approach	Normalized Time
Classical	1.0	FiP	0.36
LiT - 2 int.	1.0	LiT/FiP - 2 int.	0.33
LiT - 4 int.	1.04	LiT/FiP - 4 int.	0.35
LiT - 10 int.	1.05	LiT/FiP - 10 int.	0.36
LiT - 25 int.	1.08	LiT/FiP - 25 int.	0.38

As discussed earlier in this paper, the fixed-point iteration (FiP) approach can provide significant reductions in CPU time since the number of inner iterations for the primal solver at each time-step is reduced down to one. This computational saving can be clearly seen in Fig. 17 which presents the wall-clock times for different unsteady one-shot approaches used in this work. As can be seen, both the FiP and the proposed LiT/FiP techniques lead to considerable reductions in computational time. On the other hand, the

local-in-time (LiT) approach is shown to provide consistent reductions in memory footprint as the number of sub-intervals is increased. The results in terms of computational and memory savings are presented in Tables 1 and 2. As can be seen, the proposed LiT/FiP approach can provide up to 65% reduction in the computational time as well as more than 90% reduction in memory footprint. It must be noted that both of these significant improvements are achieved while the LiT/FiP maintains the accuracy with the quadratic programming problem converging to the same optimal solution as the classical approach.

Table 2. Normalized memory footprints for different unsteady one-shot approaches used for the optimal control of the cylinder vortex shedding via steady rotation.

One-Shot Approach	Normalized Memory Footprint	Reduction %
Classical	1.0	-
FiP	1.0	-
LiT & LiT/FiP - 2 int.	0.5	50%
LiT & LiT/FiP - 4 int.	0.25	75%
LiT & LiT/FiP - 10 int.	0.1	90%
LiT & LiT/FiP - 25 int.	0.04	96%

Next, the optimal solution for the rotational velocity amplitude is compared to the values reported by Homescu et al.[35] and Kang et al.[33] with the comparisons shown in Table 3. As can be seen, our optimal solution is in better agreement with the value reported by Kang et al.[33] which they achieved by gradually increasing the rotational velocity amplitude until the drag variations are significantly reduced.

Table 3. Comparison of the optimal rotational velocity amplitudes, Ω_{opt}, for the vortex suppression of the cylinder at Re = 100 using steady rotation.

Present	Homescu et al.[35]	Kang et al.[33]
1.8932	1.84	1.9

In order to better understand the effect of optimal rotational speed in suppressing vortex shedding and subsequently reducing the mean drag coefficient, time histories of lift and drag force coefficients are shown in Figs. 18 and 19. As mentioned earlier, the unsteady optimization problem is "hot started" by utilizing the periodic flow solution after 150 seconds. This means that the time interval of [0, 20] used as the design window is in fact the time interval between [150, 170] seconds. Once again, it can be seen that the variations in drag coefficient as well as the mean drag are significantly reduced. At the same time, and as expected, the higher rotational velocity leads to a higher lift coefficient.

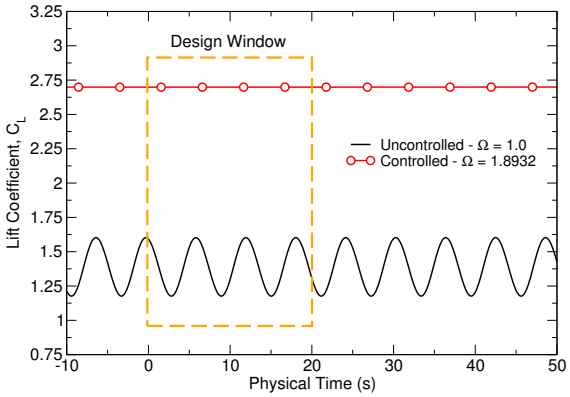


Figure 18. Time histories of the lift coefficient for the original (uncontrolled with $\Omega_{orig} = 1.0$) and optimal (controlled with $\Omega_{opt} = 1.8932$) cases of cylinder in cross-flow at Re = 100 with steady rotation.

Finally, the vorticity fields and the streamlines around the cylinder with the original and optimal surface rotational velocities are shown in Fig. 20. It can be clearly seen that the vortex street is completely suppressed

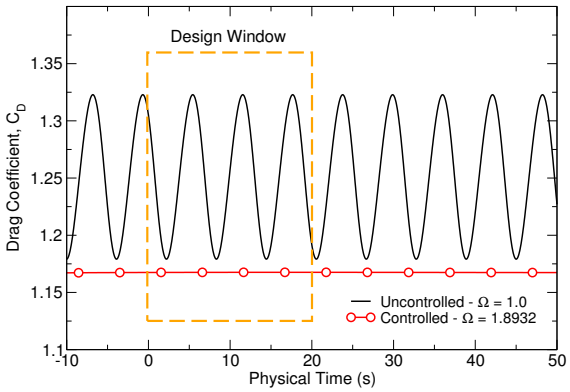
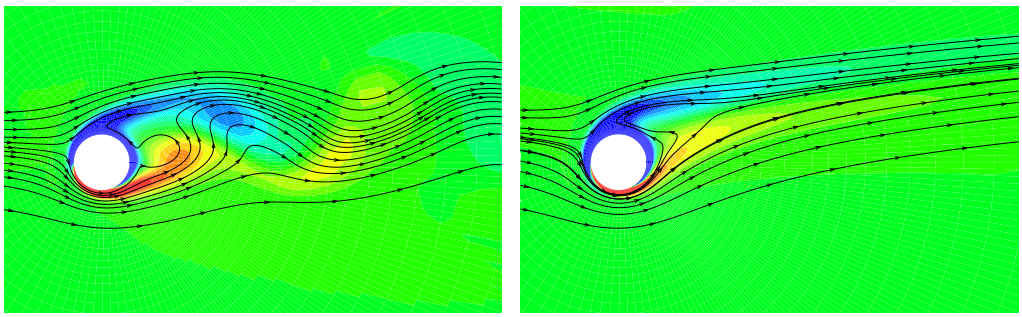


Figure 19. Time histories of the drag coefficient for the original (uncontrolled with $\Omega_{\mathbf{orig}} = 1.0$) and optimal (controlled with $\Omega_{\mathbf{opt}} = 1.8932$) cases of cylinder in cross-flow at $\mathbf{Re} = 100$ with steady rotation.

in the controlled case where the cylinder is rotating at a steady rate of $\Omega_{\text{opt}} = 1.8932$ which is found to be the optimal rotational velocity for the Reynolds number of 100 studied in this work.



(a) uncontrolled case (b) controlled case

Figure 20. Vorticity contour field and the streamlines for the original (uncontrolled with $\Omega_{\mathbf{orig}} = 1.0$) and optimal (controlled with $\Omega_{\mathbf{opt}} = 1.8932$) cases of cylinder in cross-flow at $\mathbf{Re} = 100$ with steady rotation.

2. Mean Drag Reduction via Time Periodic Rotation

Having presented the results of the unsteady design optimization pertaining the vortex suppression of a steadily rotating cylinder in cross-flow, we now shift our attention to the case where the cylinder is rotationally oscillated. As discussed before, this oscillatory motion can be defined via the rotational velocity on the surface of the cylinder defined as a time-periodic function such that

$$u_{\text{rotational}} = \Omega \, \sin(2\pi \, f \, t) \tag{58}$$

where Ω is once again the rotational velocity amplitude and f is the forcing frequency. Here, both Ω and f are chosen as the design variables with their initial values set to $\Omega_{\text{orig}} = 1.4$ and $f_{\text{orig}} = 0.3$. However, the flow-field used for "hot-starting" the design process corresponds to the natural vortex shedding of the stationary cylinder at $\text{Re} = 100$. For the present test case, a finer grid with 265×129 nodes, shown in Fig. 13(b), is utilized to better capture the flow physics in the highly unsteady wake. The design window or the time interval for the primal and adjoint calculations is taken to be $[0, 20]$ seconds, and a physical time-step of $\Delta t = 0.01$ is used.

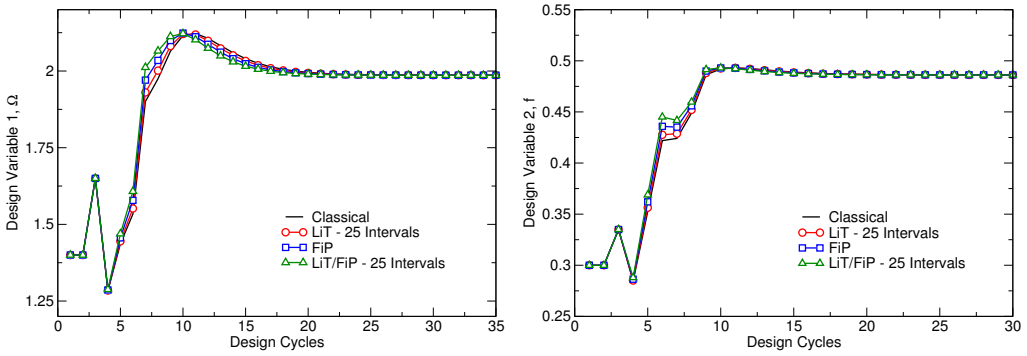
Unlike the previous test case, where vortex suppression was the main objective, the goal here is to minimize the mean drag for the oscillating cylinder. Therefore, the time-integrated quantity of interest being minimized is defined as

$$\overline{I} = \frac{1}{T} \sum_{n=1}^{N} C_D \left(\mathbf{x}^n, \mathbf{Q}^n(\mathbf{x}) \right) \Delta t \tag{59}$$

where C_D is the drag coefficient calculated based on the instantaneous flow solution and the design variables, i.e., oscillation parameters (Ω and f), according to Eq. (52). It must be noted that a similar approach has been utilized by other researchers in order to reduce the mean drag for an oscillating cylinder at various Reynolds numbers.[12,33,34,36]

In this work, the TAPENADE automatic differentiation toolbox[37] is used to calculate the sensitivities of the drag coefficient with respect to the flow solution and the design variables. Therefore, the automatically differentiated functions are called during backward-in-time adjoint solutions to obtain $\frac{\partial C_D^n}{\partial \mathbf{Q}^n}$ and $\frac{\partial C_D^n}{\partial \mathbf{x}^n}$ terms based on the current flow solutions and design variables at time level n. As discussed earlier, the sensitivities of the objective function with respect to the flow variables, i.e., $\frac{\partial I}{\partial \mathbf{Q}}$, are used on the right-hand-side of the adjoint equations (see Eqs. [39] through [42]) to propagate the solutions backward in time. At the same time, the sensitivities of the objective function with respect to the design variables, i.e., $\frac{\partial I}{\partial \mathbf{x}}$, are included in the total derivative accumulation according to Eq. (25).

Once again, the primal flow solver is run for 100 seconds with the stationary cylinder to capture the natural vortex shedding and the solution at $t = 100$ (s) is used to "hot start" the design optimization process for the [100, 120] time interval which, for brevity, will be presented as the $[0, 20]$ design window. The physical time-step is taken to be $\Delta t = 0.01$ which means 2000 time-steps are taken during the design window. Similar to the previous test case, different one-shot techniques, i.e., GiT, FiP, LiT, and LiT/FiP, are applied to solve this unsteady optimization problem. Additionally, only 25 sub-intervals are utilized for the LiT and LiT/FiP approaches which would require 80 time-steps during each sub-interval and can provide more than 90% reduction in the memory footprint.

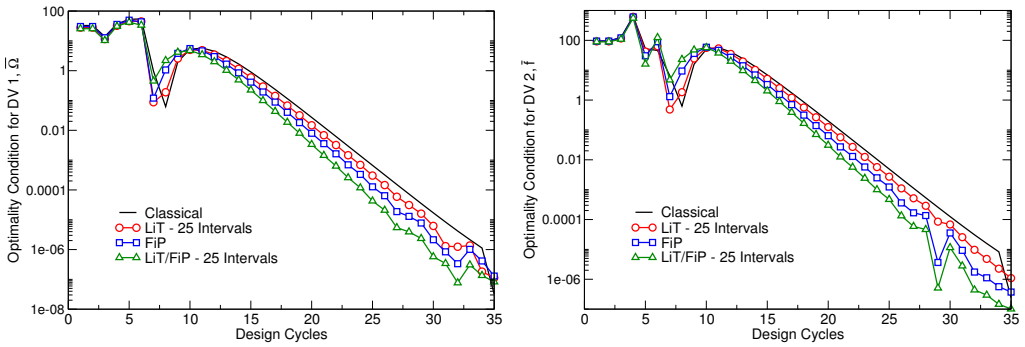


(a) Ω convergence

(b) f convergence

Figure 21. Convergence of the quadratic programming problem for the rotational velocity amplitude, Ω, and the forcing frequency, f, in the optimal periodic rotation of the cylinder in cross-flow at $\mathbf{Re} = 100$.

First, the convergence behaviors for both design variables, i.e., Ω and f, are presented in Fig. 21 using the classical (GiT), local-in-time (LiT), fixed-point iteration (FiP), as well as the proposed hybrid technique, i.e., LiT/FiP. As can be seen, there are some noticeable differences in the convergence behavior of the LiT-based techniques compared to the classical (GiT) and fixed-point iteration (FiP) approaches. This is more significant for the forcing frequency and can be once again attributed to the discontinuities in restarting the adjoint solutions in each sub-interval using an "approximate" terminal condition. Nevertheless, all cases ultimately converge to the same optimal solutions for both design variables which are $\Omega_{\text{opt}} = 1.986$ and $f_{\text{opt}} = 0.486$. At the same time, the same computational and memory savings are achieved with the FiP, LiT, and the proposed LiT/FiP approaches. It is important to once again note that the reductions in the memory footprint for the local-in-time approaches are associated with the fact that only a smaller number of time instances of the primal solution will be stored in the memory to enable the backward-in-time solution of the adjoint equations.



(a) $\frac{\partial I}{\partial \Omega}$ convergence

(b) $\frac{\partial I}{\partial \bar{f}}$ convergence

Figure 22. Optimality conditions for the quadratic programming used in determining the optimal rotational velocity amplitude and forcing frequency of the oscillating cylinder in cross-flow at Re = 100.

Next, the optimality conditions for the both design variables are presented in Fig. 22 using different unsteady one-shot techniques. Once again, it is shown that the proposed hybrid LiT/FiP approach does not negatively affect the optimality or convergence behavior of the quadratic programming. Additionally, in all cases studied here, the convergence of the optimization problem to the optimal solutions is guaranteed although the rate of convergence for this dual-variate minimization problem is comparably slower than the rate of convergence obtained in the single-variate problem presented in the previous section.

After presenting the convergence plots for different unsteady one-shot techniques, let us now focus on the optimal solutions, changes in vortex dynamics, as well as lift/drag force predictions. As shown earlier, the optimal values for the two design variables are obtained as $\Omega_{\text{opt}} = 1.986$ and $f_{\text{opt}} = 0.486$. It must be noted that all of the unsteady one-shot approaches used in this work ultimately converged to optimal solutions that were within 0.01% of each other.

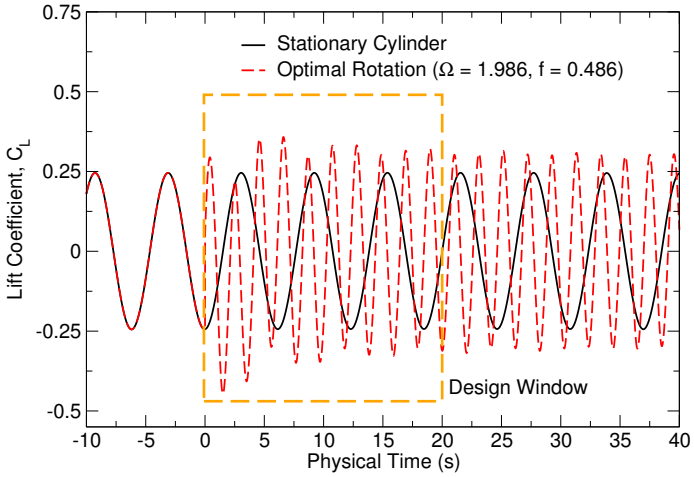


Figure 23. Time histories of the lift coefficient for the original (stationary) and optimal (oscillatory with $\Omega_{\text{opt}} = 1.986$, $f_{\text{opt}} = 0.486$) cases of cylinder in cross-flow at Re = 100.

It is worth noting that the optimal values obtained in this work are different from the optimal solutions reported by Homescu et al.[35] ($\Omega_{\text{Homescu}} \approx 6.5$ and $f_{\text{Homescu}} = 1.13$) and He et al.[34] ($\Omega_{\text{He}} \approx 6.0$ and $f_{\text{He}} = 0.75$). Interestingly, Rumpfkeil and Zingg[12] and Mehmood et al.[36] also report very different optimal values for the rotational velocity amplitude and the forcing frequency. The former work also present an in-depth study of the design space to show that there are several local optima in the lower range of forcing frequency and rotation amplitude.[12] Therefore, it can be understood that the drag minimization of the

oscillatory cylinder is a multi-modal optimization problem where, depending on the initial condition, it is possible to converge to different local optima. It would be interesting to study the multi-modality of this problem by considering various initial conditions. However, the focus of the present paper is on the use of different unsteady one-shot approaches in solving the design optimization problem with the goal of efficiency improvements in terms of CPU time and memory footprint.

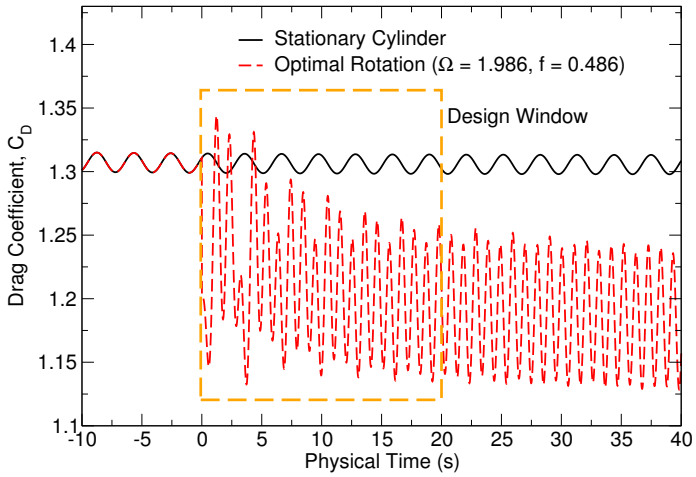


Figure 24. Time histories of the drag coefficient for the original (stationary) and optimal (oscillatory with $\Omega_{\mathbf{opt}} = 6.1388$, $f_{\mathbf{opt}} = 0.6495$) cases of cylinder in cross-flow at Re $= 100$.

Next, the lift and drag force time histories for the original and optimal designs are shown in Figs. 23 and 24. As can be seen, the optimal oscillation is capable of minimizing the mean drag value well beyond the extent of the design window, i.e., [0, 20] and a similar behavior was also observed by other researchers.[12,34,35] The detailed comparisons between the lift and drag variations as well as the reduction in the mean drag coefficient are presented in Table 4.

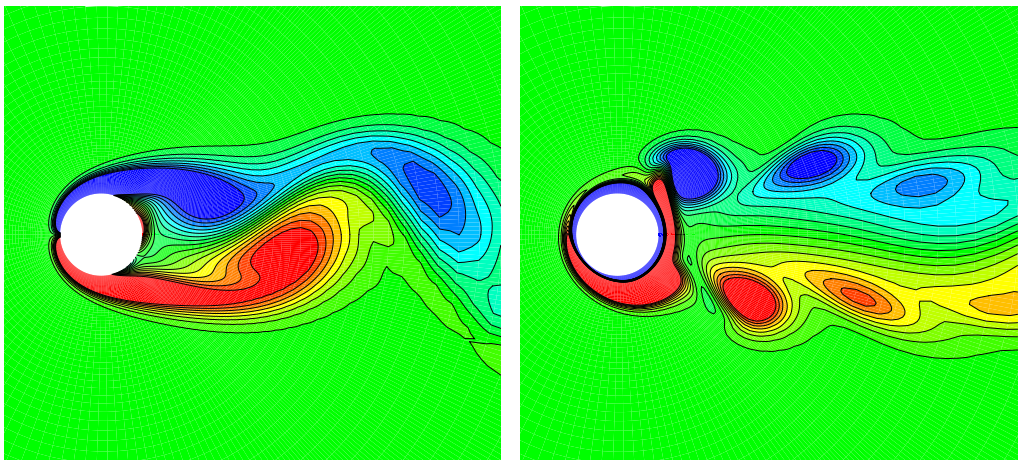
Table 4. Comparison of the lift and drag variations and mean drag reduction between the original (stationary) and optimal oscillatory rotation of the cylinder at Re $= 100$.

Cylinder Motion	C_L variation	C_D variation	C_D Mean	Mean C_D Reduction
Stationary	± 0.24	± 0.02	1.3156	-
Optimal Oscillation	± 0.29	± 0.055	1.1847	9.94%

Finally, the effect of the optimal periodic rotation of the cylinder on the vortex dynamics can be seen in the vorticity contour plots shown in Fig. 25. As can be seen, unlike the steady rotation design case presented earlier, here a complete suppression of the vortex shedding has not been achieved. However, the intensity of the vortex shedding in the wake of the cylinder oscillating with optimal rotation amplitude and forcing frequency (controlled case) is substantially reduced with the wake flow being quasi-symmetrized.[34] Nevertheless, such an optimal combination of the rotational amplitude and the forcing frequency obtained in the present work, has led to a significant reduction of the mean drag coefficient for almost 10%.

VI. Conclusions

In this paper, a new one-shot approach for unsteady adjoint-based design optimization was presented. The new method takes advantage of the local-in-time approach by breaking down the design window into a set of smaller sub-intervals which can significantly reduce the memory footprint required for storing the primal solutions needed for solving the adjoint equations backward-in-time. Additionally, a fixed-point iteration approach is utilized that can substantially improve the computational efficiency in each design cycle. The proposed hybrid technique, called LiT/FiP, is used to tackle several unsteady design optimization problems involving the lid-driven cavity flow and a circular cylinder in cross-flow. Unsteady optimization results using



(a) uncontrolled (stationary) case (b) controlled (optimal oscillation) case

Figure 25. Vorticity contour field for the original (uncontrolled with natural vortex shedding from the stationary cylinder) and optimal (controlled with $\Omega_{\mathbf{opt}} = 1.986$, $f_{\mathbf{opt}} = 0.486$) cases of cylinder in cross-flow at $\mathbf{Re} = 100$.

the LiT/FiP technique are compared to those obtained from the classical approach as well as cases with only the fixed-point iteration (FiP) and local-in-time (LiT) techniques being utilized. It has been shown that the proposed technique can provide up to 65% reduction in the computational time by utilizing the fixed-point iteration technique. At the same time, the use of the local-in-time technique with varying number of sub-intervals has shown to provide more than 95% reduction in memory footprint without sacrificing the accuracy. Through various test cases presented in this work, it is shown that, while the memory and computational savings are substantial, the proposed technique does not significantly change the convergence behavior of the optimization problem with the method ultimately converging to the same optimal solution obtained using conventional approaches.

VII. Acknowledgments

This material is based upon work supported by the National Science Foundation under grant No: CBET-1803760. The authors greatly appreciate the support provided.

References

[1]Pironneau, O., "On Optimum Design in Fluid Mechanics," *Journal of Fluid Mechanics*, Vol. 64, No. 01, 1974, pp. 97–110.

[2]Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260.

[3]Giannakoglou, K. C. and Papadimitriou, D. I., "Adjoint Methods for Shape Optimization," *Optimization and Computational Fluid Dynamics*, Springer, 2008, pp. 79–108.

[4]Giles, M. B. and Pierce, N. A., "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion*, Vol. 65, No. 3-4, 2000, pp. 393–415.

[5]Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S., *Optimization with PDE Constraints*, Vol. 23, Springer Science & Business Media, 2008.

[6]Huang, H. and Ekici, K., "A Discrete Adjoint Harmonic Balance Method for Turbomachinery Shape Optimization," *Aerospace Science and Technology*, Vol. 39, 2014, pp. 481–490.

[7]Djeddi, R. and Ekici, K., "FDOT: A Fast, Memory-Efficient and Automated Approach for Discrete Adjoint Sensitivity Analysis using the Operator Overloading Technique," *Aerospace Science and Technology*, Vol. 91, 2019, pp. 159–174.

[8]Mohammadi, B. and Pironneau, O., *Applied Shape Optimization for Fluids*, Oxford University Press, 2010.

[9]Nadarajah, S. K. and Jameson, A., "Optimum Shape Design for Unsteady Flows with Time-accurate Continuous and Discrete Adjoint Method," *AIAA Journal*, Vol. 45, No. 7, 2007, pp. 1478–1491.

[10]Nielsen, E. J., Diskin, B., and Yamaleev, N. K., "Discrete Adjoint-based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids," *AIAA Journal*, Vol. 48, No. 6, 2010, pp. 1195–1206.

[11]Carnarius, A., Thiele, F., Özkaya, E., Nemili, A., and Gauger, N. R., "Optimal Control of Unsteady Flows Using a Discrete and a Continuous Adjoint Approach," *IFIP Conference on System Modeling and Optimization*, Springer, 2011, pp. 318–327.

[12]Rumpfkeil, M. P. and Zingg, D. W., "The Optimal Control of Unsteady Flows with a Discrete Adjoint Method," *Optimization and Engineering*, Vol. 11, No. 1, 2010, pp. 5–22.

[13]Mani, K. and Mavriplis, D. J., "Unsteady Discrete Adjoint Formulation for Two-dimensional Flow Problems with Deforming Meshes," *AIAA Journal*, Vol. 46, No. 6, 2008, pp. 1351–1364.

[14]Griewank, A. and Walther, A., "Algorithm 799: Revolve: an Implementation of Checkpointing for the Reverse or Adjoint Mode of Computational Differentiation," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 26, No. 1, 2000, pp. 19–45.

[15]Griewank, A., "Achieving Logarithmic Growth of Temporal and Spatial Complexity in Reverse Automatic Differentiation," *Optimization Methods and Software*, Vol. 1, No. 1, 1992, pp. 35–54.

[16]Wang, Q., Moin, P., and Iaccarino, G., "Minimal Repetition Dynamic Checkpointing Algorithm for Unsteady Adjoint Calculation," *SIAM Journal on Scientific Computing*, Vol. 31, No. 4, 2009, pp. 2549–2567.

[17]Hückelheim, J. C. and Müller, J.-D., "Checkpointing with Time Gaps for Unsteady Adjoint CFD," *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Springer, 2019, pp. 117–130.

[18]Yamaleev, N. K., Diskin, B., and Nielsen, E. J., "Local-in-time Adjoint-based Method for Design Optimization of Unsteady Flows," *Journal of Computational Physics*, Vol. 229, No. 14, 2010, pp. 5394–5407.

[19]Yamaleev, N., Diskin, B., and Nielsen, E., "Adjoint-based Methodology for Time-dependent Optimization," AIAA Paper 2008-5857, 2008.

[20]Beran, P., Stanford, B., and Kurdi, M., "Sensitivity Analysis for Optimization of Dynamic Systems with Reduced Order Modeling," AIAA Paper 2010-1503, 2010.

[21]Biegler, L. T., Ghattas, O., Heinkenschloss, M., and van Bloemen Waanders, B., "Large-scale PDE-constrained Optimization: An Introduction," *Large-Scale PDE-Constrained Optimization*, Springer, 2003, pp. 3–13.

[22]Özkaya, E., *One-shot Methods for Aerodynamic Shape Optimization*, Ph.D. thesis, RWTH Aachen University, 2014.

[23]Ta'asan, S., "Pseudo-time Methods for Constrained Optimization Problems Governed by PDE." Tech. rep., Institute for Computer Applications in Science and Engineering, Hampton, VA, 1995.

[24]Hazra, S., Schulz, V., Brezillon, J., and Gauger, N., "Aerodynamic Shape Optimization Using Simultaneous Pseudo-timestepping," *Journal of Computational Physics*, Vol. 204, No. 1, 2005, pp. 46–64.

[25]Hazra, S. B., "Aerodynamic Shape Optimization Using Simultaneous Pseudo-Time-Stepping," *Large-Scale PDE-Constrained Optimization in Applications*, Springer, 2010, pp. 81–104.

[26]Günther, S., Gauger, N. R., and Wang, Q., "Simultaneous Single-step One-shot Optimization with Unsteady PDEs," *Journal of Computational and Applied Mathematics*, Vol. 294, 2016, pp. 12–22.

[27]Günther, S., *Simultaneous Optimization with Unsteady Partial Differential Equations*, Ph.D. thesis, RWTH Aachen University, 2017.

[28]Garmann, D. J., "Compact finite-differencing and filtering procedure applied to the incompressible Navier-Stokes equations," *AIAA journal*, Vol. 51, No. 9, 2013, pp. 2241–2251.

[29]Von Karman, T., "Über den Mechanismus des Widerstandes, den ein bewegter Körper in einer Flüssigkeit erfährt," *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, Vol. 1911, 1911, pp. 509–517.

[30]Williamson, C. H., "Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers," *Journal of Fluid Mechanics*, Vol. 206, 1989, pp. 579–627.

[31]Henderson, R. D., "Nonlinear dynamics and pattern formation in turbulent wake transition," *Journal of fluid mechanics*, Vol. 352, 1997, pp. 65–112.

[32]Tokumaru, P. and Dimotakis, P., "Rotary oscillation control of a cylinder wake," *Journal of Fluid Mechanics*, Vol. 224, 1991, pp. 77–90.

[33]Kang, S., Choi, H., and Lee, S., "Laminar flow past a rotating circular cylinder," *Physics of Fluids*, Vol. 11, No. 11, 1999, pp. 3312–3321.

[34]He, J.-W., Glowinski, R., Metcalfe, R., Nordlander, A., and Periaux, J., "Active control and drag optimization for flow past a circular cylinder: I. Oscillatory cylinder rotation," *Journal of Computational Physics*, Vol. 163, No. 1, 2000, pp. 83–117.

[35]Homescu, C., Navon, I., and Li, Z., "Suppression of vortex shedding for flow around a circular cylinder using optimal control," *International Journal for Numerical Methods in Fluids*, Vol. 38, No. 1, 2002, pp. 43–69.

[36]Mehmood, A., Hajj, M., Akhtar, I., Ghommem, M., Watson, L., and Lux, T., "Optimized drag reduction and wake dynamics associated with rotational oscillations of a circular cylinder," *Contemporary Engineering Sciences*, Vol. 11, No. 97, 2018.

[37]Hascoët, L., "TAPENADE: a tool for Automatic Differentiation of programs," *Proceedings of 4th European Congress on Computational Methods, ECCOMAS*, 2004, pp. 1–14.