



Efficient One-Shot Technique for Adjoint-Based Unsteady Optimization

Reza Djeddi[✉] and Kivanc Ekici[†]

University of Tennessee, Knoxville, Tennessee 37996

<https://doi.org/10.2514/1.J.060142>

A computationally efficient one-shot approach with a low memory footprint is presented for unsteady 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 in which 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 crossflow 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 optimization problem converges to the same optimal solution obtained using a conventional approach.

Nomenclature

| | |
|---------------|--|
| B | = preconditioning or Hessian matrix |
| f | = rotational frequency |
| l | = objective function |
| J | = Jacobian of transformation from physical to computational space |
| K | = excitation parameter |
| \mathcal{L} | = Lagrangian functional |
| M, N | = number of subintervals and time steps, respectively |
| Q | = vector of primal or state solutions |
| R | = vector of residuals of the primal solver |
| r | = radius |
| Str | = Strouhal number |
| T | = time span or the length of the design interval |
| t | = time |
| u, v | = Cartesian components of the velocity vector |
| x | = vector of design variables |
| λ | = vector of adjoint or costate solutions |
| ξ, η | = parametric coordinates corresponding to the (x, y) Cartesian coordinates |
| ψ | = stream function |
| ω | = vorticity |
| Ω | = rotational speed of cylinder |

I. Introduction

ADJOINT-BASED optimization has become a standard technique in aerospace engineering. In this approach, the adjoint equations are derived by adjoining the linearized primal or state equations. These equations are then solved to obtain the costate 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 [1–3], which is the focus of the present work. Normally, it is assumed that each design would determine a unique solution for the 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 or a reduced space in which the constraint for the existence of the state solution is treated implicitly. Alternatively, a full-space method has been proposed in which the optimality conditions are solved for the primal, adjoint, and design variables all at the same time [4], thus treating the constraint for the existence of the state solution explicitly. Since the primal and adjoint solutions are directly incorporated into the design optimization problem, this monolithic 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 [5,6]. Additionally, Ta'asan [10] and Hazra et al. [11,12] proposed a simultaneous pseudo time-stepping approach to solve the three equations (primal, adjoint, and design) derived from the optimality conditions.

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 [13–26]. 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 backward 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 [27,28].

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 [29] and Wang et al. [30] that was based on a divide-and-conquer strategy proposed by Griewank [31]. In this approach, only a few snapshots of the primal solutions are stored at certain time instances (or checkpoints) during the optimization time interval. Recently, Hückelheim and Müller [32] have proposed a new “gappy” checkpointing approach that can offer greater memory reduction with

Presented as Paper 2020-3129 at the AIAA AVIATION Forum, June 15–19, 2020 (Virtual Event); received 22 September 2020; revision received 5 February 2021; accepted for publication 21 February 2021; published online 23 June 2021. Copyright © 2021 by the authors. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 1533-385X to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*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.

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 [29].

In general, unsteady design problems aim to minimize a time-integrated objective function over a desired time interval using a technique often referred to as global in time [33,34]. In an approach similar to checkpointing, Yamaleev et al. [35], proposed dividing this time interval into several smaller subintervals in which 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 [36]. Beran et al. [37] have also proposed the use of a data-reduction approach based on the proper orthogonal decomposition method for memory reduction. More recently, Günther et al. [38] 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.

The present work aims at combining the memory efficiency of the local-in-time (LiT) approach and the computational robustness of the fixed-point iteration (FiP) approach in developing a highly efficient adjoint-based optimization technique for unsteady design problems. The proposed technique is similar to the SAND approach with piggyback iterations in which the primal and the adjoint equations are solved simultaneously by freezing the design variables [8]. 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. The combination of the local-in-time and fixed-point iteration approaches, reported for the first time in the literature, leads to robust time-accurate adjoint-based optimization for two-dimensional incompressible laminar flow cases considered in this work. 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 used 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 Optimization

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

$$\min I(\mathbf{x}, \mathbf{Q}(x)) \quad \text{subject to } \mathbf{R}(\mathbf{x}, \mathbf{Q}(x)) = 0 \quad (1)$$

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. Using the method of Lagrange multipliers, the minimization problem in Eq. (1) can be reformulated as an unconstrained minimization of a Lagrangian functional such that

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

where λ is the vector of adjoint or costate solutions. An optimal solution exists for the minimization of the augmented Lagrangian functional if the Karush–Kuhn–Tucker (KKT) optimality conditions are satisfied such that

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Q}} = 0 \rightarrow \frac{\partial \mathcal{L}}{\partial \mathbf{Q}} = \frac{\partial I}{\partial \mathbf{Q}} + \lambda^T \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} = 0 \rightarrow \nabla_{\mathbf{Q}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \lambda) = 0 \quad (3)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = 0 \rightarrow \frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial I}{\partial \mathbf{x}} + \lambda^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}} = 0 \rightarrow \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \lambda) = 0 \quad (4)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \rightarrow \frac{\partial \mathcal{L}}{\partial \lambda} = \mathbf{R}(\mathbf{x}, \mathbf{Q}(x)) = 0 \rightarrow \mathbf{R}(\mathbf{x}, \mathbf{Q}(x)) = 0 \quad (5)$$

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

$$\begin{bmatrix} \frac{\partial^2 I}{\partial \mathbf{Q}^2} & \frac{\partial^2 I}{\partial \mathbf{Q} \partial \mathbf{x}} & \frac{\partial \mathbf{R}^T}{\partial \mathbf{Q}} \\ \frac{\partial^2 I}{\partial \mathbf{Q} \partial \mathbf{x}} & \frac{\partial^2 I}{\partial \mathbf{x}^2} & \frac{\partial \mathbf{R}^T}{\partial \mathbf{x}} \\ \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} & \frac{\partial \mathbf{R}}{\partial \mathbf{x}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{Q} \\ \Delta \mathbf{x} \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{Q}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \lambda) \\ -\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \lambda) \\ -\mathbf{R}(\mathbf{x}, \mathbf{Q}(x)) \end{bmatrix} \quad (6)$$

to determine the updates for the state, costate, and design solutions all in the same fully coupled framework. To reduce the computational cost of solving Eq. (6), Hazra [12] suggested that terms $\partial^2 I / \partial \mathbf{Q}^2$ and $\partial^2 I / \partial \mathbf{Q} \partial \mathbf{x}$ can be ignored with little to no effect on accuracy. Therefore, the simplified system of one-shot governing equations can be written as

$$\begin{bmatrix} 0 & 0 & \frac{\partial \mathbf{R}^T}{\partial \mathbf{Q}} \\ 0 & \frac{\partial^2 I}{\partial \mathbf{x}^2} & \frac{\partial \mathbf{R}^T}{\partial \mathbf{x}} \\ \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} & \frac{\partial \mathbf{R}}{\partial \mathbf{x}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{Q} \\ \Delta \mathbf{x} \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{Q}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \lambda) \\ -\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{Q}, \lambda) \\ -\mathbf{R}(\mathbf{x}, \mathbf{Q}(x)) \end{bmatrix} \quad (7)$$

In the preceding equation, the Hessian of the objective function $\partial^2 I / \partial \mathbf{x}^2$ used for the quadratic programming can be approximated using a few different approaches. According to Hazra et al. [11,12] and Özkaya [8], the Hessian can be simply replaced by 1) a diagonal matrix for a linear steepest descent optimization or 2) a BFGS-like (named after the work of Broyden-Fletcher-Goldfarb-Shanno [39]) approximation of the Hessian based on the gradient information $\partial I / \partial \mathbf{x}$ from two subsequent design cycles for a superlinear convergence to the optimal solution. Next, the one-shot method for unsteady adjoint problems is discussed.

III. Unsteady Adjoint-Based Optimization

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

$$\bar{I} = \frac{1}{T} \int_0^T I(\mathbf{x}, \mathbf{Q}(t, \mathbf{x})) dt \quad (8)$$

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

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{R}(\mathbf{x}, \mathbf{Q}(t, \mathbf{x})) = 0 \quad \forall t \in (0, T] \quad (9)$$

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

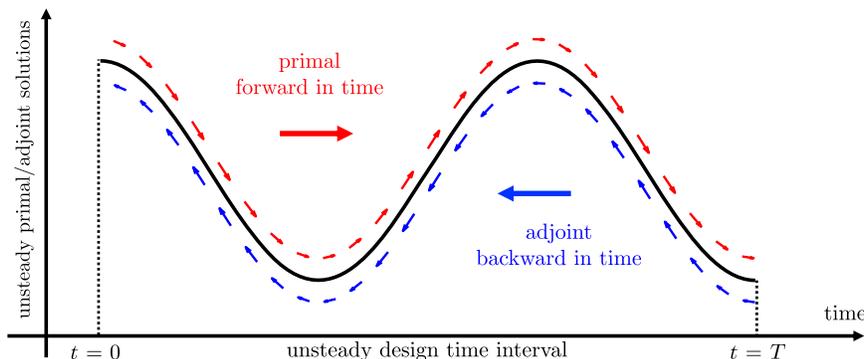


Fig. 1 Forward-in-time procedure to obtain the primal solutions followed by the backward-in-time solution for the adjoints in a typical unsteady design problem over the time interval $[0, T]$.

$$\begin{aligned} \min \bar{I} &= \frac{1}{T} \int_0^T I(x, \mathbf{Q}(t, x)) dt \\ \text{subject to } \frac{\partial \mathbf{Q}}{\partial t} + \mathbf{R}(x, \mathbf{Q}(t, x)) &= 0 \quad \forall t \in (0, T) \\ \text{with } \mathbf{Q} &= \mathbf{Q}^{\text{init}} \text{ for } t = 0 \end{aligned} \quad (10)$$

where the second constraint is simply the initial condition for the unsteady primal solution. The semidiscretized unsteady governing equations shown in Eq. (9) are solved using classical dual-time-stepping schemes where, depending on the order of the temporal discretization, different numbers of primal solutions from earlier time steps are used. As an example, the first-order backward difference (BDF1) and the second-order backward difference (BDF2) schemes are written as

$$\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}(x, \mathbf{Q}^n(x)) = 0 \quad (11)$$

$$\frac{3\mathbf{Q}^n - 4\mathbf{Q}^{n-1} + \mathbf{Q}^{n-2}}{2\Delta t} + \mathbf{R}(x, \mathbf{Q}^n(x)) = 0 \quad (12)$$

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

$$\bar{I} = \frac{1}{T} \int_0^T I(x, \mathbf{Q}(t, x)) dt \approx \frac{1}{T} \sum_{n=1}^N I(x, \mathbf{Q}^n(x)) \Delta t \quad (13)$$

For the sake of brevity, let us only focus on the BDF1 approach and rewrite the minimization problem for the objective function [Eq. (13)] constrained by the discretized governing equations [Eq. (11)] as well as the initial condition.

$$\begin{aligned} \min \bar{I} &= \frac{1}{T} \sum_{n=1}^N I(x, \mathbf{Q}^n(x)) \Delta t \\ \text{subject to } \frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}(x, \mathbf{Q}^n(x)) &= 0 \text{ for } n = 2, 3, \dots, N \\ \text{with } \mathbf{Q}^1 &= \mathbf{Q}^{\text{init}} \end{aligned} \quad (14)$$

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

$$\begin{aligned} \mathcal{L}(x, \mathbf{Q}, \boldsymbol{\lambda}) &= \frac{1}{T} \left[\sum_{n=1}^N I(x, \mathbf{Q}^n(x)) \Delta t \right. \\ &\quad \left. + \sum_{n=2}^N [\boldsymbol{\lambda}^n]^T \left(\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n \right) \Delta t + [\boldsymbol{\lambda}^1]^T (\mathbf{Q}^1 - \mathbf{Q}^{\text{init}}) \right] \end{aligned} \quad (15)$$

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

$$\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n = 0 \quad \text{for } n = 2, 3, \dots, N \quad (16)$$

$$\mathbf{Q}^1 - \mathbf{Q}^{\text{init}} = 0 \rightarrow \mathbf{Q}^1 = \mathbf{Q}^{\text{init}} \quad (17)$$

which are also the constraints of the unsteady minimization problem. Similarly, the first optimality condition (i.e., $\partial \mathcal{L} / \partial \mathbf{Q} = 0$) will result in the adjoint or costate equations given as

$$\frac{\boldsymbol{\lambda}^N}{\Delta t} + \left[\frac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N} \right]^T \boldsymbol{\lambda}^N = - \frac{\partial I^N}{\partial \mathbf{Q}^N} \quad \text{terminal condition, } (n = N) \quad (18)$$

$$\frac{\boldsymbol{\lambda}^n - \boldsymbol{\lambda}^{n+1}}{\Delta t} + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n} \right]^T \boldsymbol{\lambda}^n = - \frac{\partial I^n}{\partial \mathbf{Q}^n} \quad \text{intermediate states,} \\ (n = 2, \dots, N-1) \quad (19)$$

$$\frac{\boldsymbol{\lambda}^1 - \boldsymbol{\lambda}^2}{\Delta t} = - \frac{\partial I^1}{\partial \mathbf{Q}^1} \quad \text{initial condition, } (n = 1) \quad (20)$$

As seen from Eqs. (16–20), 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 ($n = 1, \dots, N$) 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 ($n = N, \dots, 1$), as depicted in Fig. 1. It must be noted that the right-hand-side vector in Eqs. (18–20) can be evaluated by automatically differentiating (AD) the subroutine that calculates the instantaneous objective function I^n as a function of the flow solution \mathbf{Q}^n . This can be done using source code transformation AD tools like TAPENADE[‡] [40] or operator overloading tools like Fast automatic Differentiation based on the Operator-Overloading Technique (FDOT) [6], which was developed recently by the authors. Also note that the primal and adjoint

[‡]TAPENADE is an Automatic Differentiation Engine [40] that can perform tangent, adjoint, and hybrid modes of AD based on the source-code-transformation approach.

equations using the BDF2 time-marching scheme can be derived in a similar approach.

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

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\mathbf{x}} &= \frac{1}{T} \sum_{n=1}^N \frac{\partial I^n}{\partial\mathbf{x}} \Delta t \\ &+ \dots \underbrace{\left[\frac{\partial\mathbf{Q}^N}{\partial\mathbf{x}} \right]^T \left(\frac{\lambda^N}{\Delta t} + \left[\frac{\partial\mathbf{R}^N}{\partial\mathbf{Q}^N} \right]^T \lambda^N + \frac{\partial I^N}{\partial\mathbf{Q}^N} \right)}_{=0 \text{ Eq.(18)}} \Delta t \\ &+ \dots \sum_{n=2}^N \underbrace{\left[\frac{\partial\mathbf{Q}^n}{\partial\mathbf{x}} \right]^T \left(\frac{\lambda^n - \lambda^{n+1}}{\Delta t} + \left[\frac{\partial\mathbf{R}^n}{\partial\mathbf{Q}^n} \right]^T \lambda^n + \frac{\partial I^n}{\partial\mathbf{Q}^n} \right)}_{=0 \text{ Eq.(19)}} \Delta t \\ &+ \dots \underbrace{\left[\frac{\partial\mathbf{Q}^1}{\partial\mathbf{x}} \right]^T \left(\frac{\lambda^1 - \lambda^2}{\Delta t} + \frac{\partial I^1}{\partial\mathbf{Q}^1} \right)}_{=0 \text{ Eq.(20)}} \Delta t - \left[\frac{\partial\mathbf{Q}^{\text{init}}}{\partial\mathbf{x}} \right]^T \lambda^{(1)} \\ &+ \dots \sum_{n=2}^N \left[\frac{\partial\mathbf{R}^n}{\partial\mathbf{x}} \right]^T \lambda^n \Delta t \end{aligned} \quad (21)$$

From Eqs. (18–20), the three terms inside the parentheses in the preceding equation vanish. Therefore, the simplified form of the second optimality condition reads

$$\frac{\partial\mathcal{L}}{\partial\mathbf{x}} = \frac{1}{T} \sum_{n=1}^N \frac{\partial I^n}{\partial\mathbf{x}} \Delta t + \sum_{n=2}^N \left[\frac{\partial\mathbf{R}^n}{\partial\mathbf{x}} \right]^T \lambda^n \Delta t - \left[\frac{\partial\mathbf{Q}^{\text{init}}}{\partial\mathbf{x}} \right]^T \lambda^1 \quad (22)$$

which provides 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. (16) and (17), then the Lagrangian functional would be the same as the original time-averaged objective function \bar{I} defined in Eq. (8):

$$\frac{\partial\mathcal{L}}{\partial\mathbf{x}} = \frac{\partial\bar{I}}{\partial\mathbf{x}} = \frac{1}{T} \sum_{n=1}^N \frac{\partial I^n}{\partial\mathbf{x}} \Delta t + \sum_{n=2}^N \left[\frac{\partial\mathbf{R}^n}{\partial\mathbf{x}} \right]^T \lambda^n \Delta t - \left[\frac{\partial\mathbf{Q}^{\text{init}}}{\partial\mathbf{x}} \right]^T \lambda^1 \quad (23)$$

This original one-shot approach for unsteady design optimization is referred to in this work as the global-in-time (GiT) technique.

A. Local-in-Time Approach for Memory Efficiency

In order to motivate the LiT approach developed in this work, let us first describe the piggyback-iterations technique. Initially, the primal

equations are solved forward in time, and the solutions at each time step are stored in memory. Next, the adjoint equations are solved backward in time using the stored primal solutions, followed by the calculation of the total sensitivity of the time-integrated objective function [Eq. (23)] based on the calculated primal and adjoint solutions. Using a sequential quadratic programming (SQP) approach, the design variables can be updated by

$$\mathbf{x}^{c+1} = \mathbf{x}^c - \mathbf{B}_c^{-1} \left(\frac{\partial\bar{I}}{\partial\mathbf{x}} \right)_c \quad (24)$$

where c is the design cycle number and \mathbf{B} is a preconditioning matrix that approximates the Hessian of the augmented Lagrangian [see Eq. (15)] or, in the case of the piggyback iterations, the time-integrated objective given in Eq. (8) with respect to the design variables. As discussed earlier, following the technique originally proposed by Yamaleev et al. [55], the time span $[0, T]$ is divided into M smaller subintervals where

$$\text{interval } m: (m-1) \times \left(\frac{T}{M} \right) \leq t \leq m \times \left(\frac{T}{M} \right) \quad \text{for } m = 1, \dots, M \quad (25)$$

Therefore, the primal equations [Eqs. (16) and (17)] and adjoint equations [Eqs. (18–20)] can be solved over each subinterval instead of the entire time period, as shown in Fig. 2. This approach leads to a $100 \times [N - (N/M)]/N\%$ reduction in the memory footprint for storing primal solutions, which increases if more subintervals are used over the time period.

B. Fixed-Point Iteration for Improved Computational Efficiency

As discussed before, the idea of fixed-point iterations for evaluating the adjoints is not limited to the unsteady design problems and has also been used for steady applications [6]. Additionally, a one-shot optimization approach based on the fixed-point iterations for unsteady design problems was originally proposed by Günther et al. [38,41]. To motivate this approach, let us once again consider the minimization of a time-averaged objective function according to Eq. (10) where the unsteady partial differential equation (PDE) constraint is temporally discretized using a BDF1 scheme. The constraint for this minimization problem can be rewritten in the following form:

$$\begin{aligned} \frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}(\mathbf{x}, \mathbf{Q}^n(x)) &= \mathbf{R}'(\mathbf{x}, \mathbf{Q}^n(x), \mathbf{Q}^{n-1}(x)) = 0 \\ \text{for } n &= 2, 3, \dots, N \end{aligned} \quad (26)$$

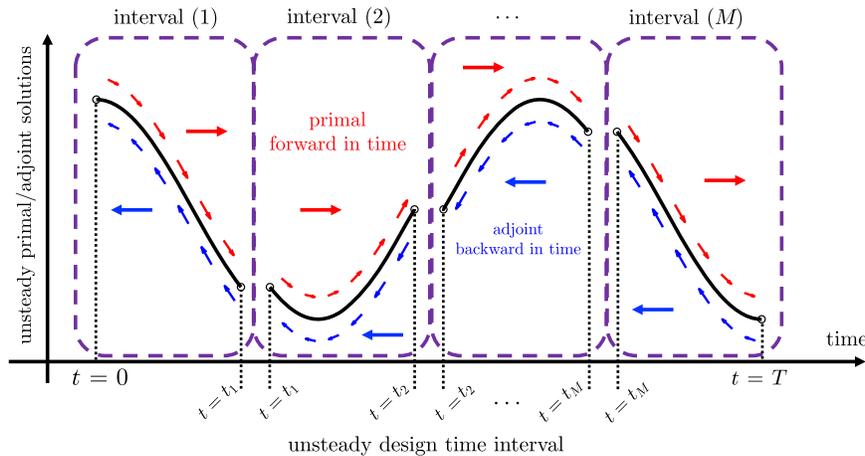


Fig. 2 Design time interval $[0, T]$ is divided into M subintervals for the local-in-time approach. Here, $t_1 = (T/M)$, $t_2 = 2(T/M)$, and $t_M = (M-1)(T/M)$.

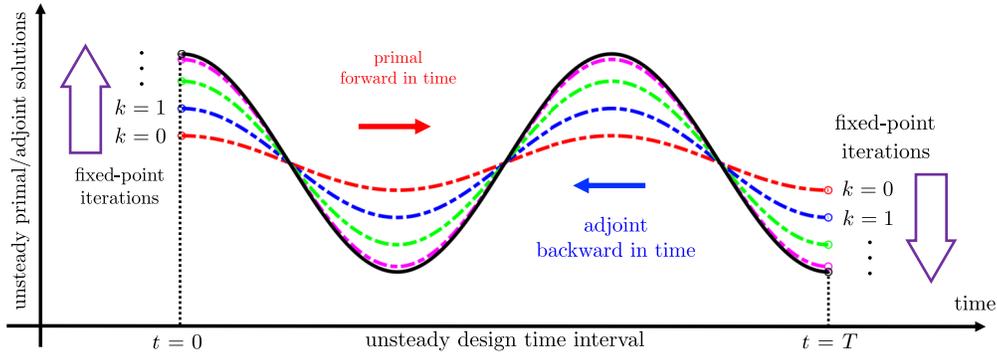


Fig. 3 Progressing unconverged primal solutions forward in time for each fixed-point iteration over the entire time interval $[0, T]$, followed by direct solutions of the linear adjoint equations in the FiP approach.

where R' is the total residual of the unsteady primal equations. In the approach proposed by Günther et al. [38], the unsteady primal equations [Eq. (26)] are solved via a fixed-point iteration so that

$$\begin{aligned} &\text{loop 1 : for } n = 1, \dots, N \\ &\quad \text{loop 2 : for } k = 1, \dots \\ &\quad \quad Q_{k+1}^n = F_p(x, Q_k^n(x), Q^{n-1}(x)) \end{aligned} \quad (27)$$

where F_p is the primal fixed-point iterator designed in such a way that

$$\begin{aligned} &\text{for } k \rightarrow \infty : Q_{k+1}^n = F_p(x, Q_k^n(x), Q^{n-1}(x)) = Q_*^n \\ &\quad \text{where } R'(x, Q_*^n(x), Q^{n-1}(x)) = 0 \end{aligned} \quad (28)$$

It is worth noting that in Eq. (27), the first loop is based on physical time stepping to advance the solution forward in time, while the second loop is based on fixed-point iterations (or inner iterations) that advance the solution in pseudotime. The idea of one-shot approach based on fixed-point iteration is to simply swap the two iteration loops in Eq. (27), which means that the primal solutions are advanced in physical time, even though the inner iterations have not converged yet. Günther et al. [38,41] mathematically proved that the convergence of the fixed-point iteration is guaranteed as long as Eq. (27) is convergent.

While the idea of using a fixed-point iteration for solving the adjoint equations was proposed in the literature [4], this work focuses on the direct solution of the adjoint equations. Since the adjoint equations are linearized, their convergence is guaranteed for a convergent primal solution. In the first stage of the FiP approach, a single step of the primal fixed-point iteration is solved for all physical time steps while storing the primal solutions in the memory. This is followed by solving the adjoint equations [Eqs. (18–20)] backward in time. The primal and adjoint solutions are then used to update the

design variables according to the SQP formulation [Eq. (24)], and the process is repeated until the optimality conditions are satisfied. The schematic of the FiP approach is provided in Fig. 3 for clarity. As will be demonstrated in the Sec. V the use of 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 may increase the number of design cycles required to reach an optimal solution. However, the savings in computational cost are significant enough to make this an efficient and robust unsteady adjoint-based optimization technique.

C. New Hybrid Approach

Following the two approaches described earlier, the main objective of this work is to couple the local-in-time gradient evaluation for memory efficiency and fixed-point iterations for improved computational efficiency. Based on our state-of-the-art knowledge, this is the first reported work in the literature that successfully merges these two techniques. The idea, referred to as the LiT/FiP approach, is straightforward in the sense that each time-advance loop based on the fixed-point iteration is also divided into smaller subintervals 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. However, unlike the checkpointing approach, the backward-in-time adjoint solutions for each subinterval are initialized using terminal conditions from the previous design cycle without the need for the recalculations of the missing primal solutions. Therefore, the proposed hybrid approach can provide both computational and memory savings at the same time without sacrificing one for the other.

To better understand the proposed hybrid approach for unsteady one-shot design, the entire procedure is described in Algorithm 1. As can be seen, the proposed technique involves two main loops, where the outer loop searches for the optimal design while also repeating

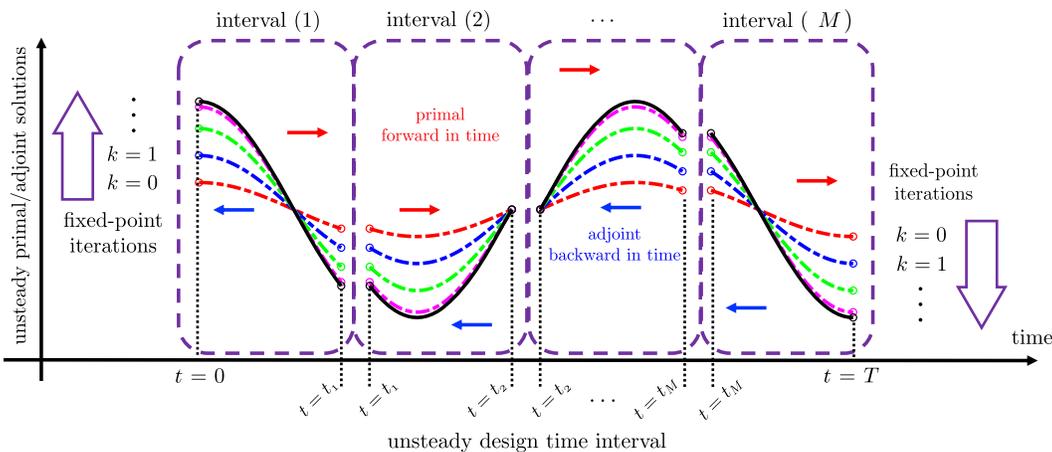


Fig. 4 The proposed hybrid approach for unsteady design in interval $[0, T]$ with M subintervals based on the fixed-point approach.

the fixed-point iterations that are performed at each design cycle. On the other hand, the inner loop follows the subintervals for the local-in-time approach where, in each subinterval, the forward and backward time-marching loops are performed that lead to the accumulation of the total derivatives of the augmented Lagrangian functional with respect to the vector of design variables. Ultimately, this total derivative is used in a Newton's update formula that advances the design variables toward the optimality condition. This process is continued until all optimality (KKT) conditions are satisfied, which at the same time would result in the convergence of the fixed-point iterations for the primal flow equations.

Once again, it must be noted that the adjoint equations presented in this section are all based on the BDF1 time-marching scheme. However, in this work, a second-order BDF2 dual-time-stepping approach is used, resulting in higher temporal accuracy in capturing unsteady flow features with rapid transience. Therefore, the adjoint equations for the BDF2 scheme will be described in the following section, and they simply replace Eqs. (18–20) described in this section as well as in Algorithm 1 for updating the adjoint solution λ in each design window.

Algorithm 1: New hybrid LiT/FiP approach

Result: Optimal solution \mathbf{x}^{opt} for unsteady design in interval $t = [0, T]$ with M subintervals

“initialize” or “hot-start” flow

while optimal solution has not reached **do**

start of the first subinterval, $m = 1$

while $m \leq M$ **do**

start forward-in-time pass with $n = (m - 1) \times (\frac{T}{M})$

while $n \leq m \times (\frac{T}{M})$ **do**

update primal solution \mathbf{Q}_{k+1}^n using a single fixed-point iteration [Eq. (23)]

write flow solution to disk

increment time step ($n = n + 1$)

end

start backward-in-time pass with $n = (m) \times (\frac{T}{M})$

set terminal condition [Eq. (18)] based on λ values from previous design cycle

while $n \geq (m - 1) \times (\frac{T}{M})$ **do**

read flow solution from disk

update adjoint solution λ using Eqs. (18–20)

decrement time step ($n = n - 1$)

end

add contributions to $\frac{\partial \mathcal{L}}{\partial \mathbf{x}}$ [Eq. (23)]

goto next interval ($m = m + 1$)

end

update design vector \mathbf{x} [Eq. (24)] using the piggybacked total derivative

check optimality conditions [Eqs. (3–5)]

goto next design cycle

end

IV. Governing Equations for the Primal and Adjoint CFD Solver

Before presenting the unsteady design optimization results, the governing equations for the primal and adjoint CFD solvers are provided in this section. An in-house code, GENESIS, solves the stream function and vorticity form of the two-dimensional incompressible Navier–Stokes equations written in a generalized coordinate system through a transformation from Cartesian coordinates (x, y) to computational coordinates (ξ, η) such that [42]

$$\frac{\partial \omega}{\partial t} + J \left(\frac{\partial \psi}{\partial \eta} \frac{\partial \omega}{\partial \xi} - \frac{\partial \psi}{\partial \xi} \frac{\partial \omega}{\partial \eta} \right) - \frac{1}{Re} \left(\alpha \frac{\partial^2 \omega}{\partial \xi^2} + 2\gamma \frac{\partial^2 \omega}{\partial \xi \partial \eta} + \beta \frac{\partial^2 \omega}{\partial \eta^2} + P \frac{\partial \omega}{\partial \xi} + Q \frac{\partial \omega}{\partial \eta} \right) = 0 \quad (29)$$

$$\alpha \frac{\partial^2 \psi}{\partial \xi^2} + 2\gamma \frac{\partial^2 \psi}{\partial \xi \partial \eta} + \beta \frac{\partial^2 \psi}{\partial \eta^2} + P \frac{\partial \psi}{\partial \xi} + Q \frac{\partial \psi}{\partial \eta} = -\omega \quad (30)$$

where $J = \partial(\xi, \eta) / \partial(x, y) = (x_\xi y_\eta - x_\eta y_\xi)^{-1}$ is the Jacobian of the transformation, and the auxiliary coefficients (i.e., α, β, γ, P , and Q) are described in Ref. [42]. GENESIS uses second-order finite differences for the spatial derivatives, resulting in a semidiscrete form written as

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{R}(\mathbf{Q}) = 0 \quad (31)$$

where \mathbf{Q} is the vector of flow variables defined as $\mathbf{Q} = [\psi \ \omega]^T$. Here, the residual of the primal CFD solver $\mathbf{R}(\mathbf{Q})$ includes the linearized form of the spatially discretized governing equations [i.e., $\mathbf{R}(\mathbf{Q}) = \mathbf{A} \cdot \Delta \mathbf{Q} - \mathbf{S}(\mathbf{Q})$], where \mathbf{A} is the Jacobian matrix that can be analytically derived for a second-order linearization in space. Note that Eq. (31) is the constraint PDE for the minimization problem as described earlier via Eq. (9). It is important to note that while the system of governing equations is written in the form of Eq. (31), 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. (18–20). Therefore, based on the KKT optimality condition for the state solution, the costate equations for the adjoint vector $\lambda = [\lambda_\psi \ \lambda_\omega]^T$ are derived as

$$\frac{\lambda_\omega^N}{\Delta t} + \left[\frac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N} \right]^T \lambda_\omega^N = -\frac{\partial I^N}{\partial \mathbf{Q}^N} \quad \text{terminal condition, } (n = N) \quad (32)$$

$$\frac{\lambda_\omega^{N-1} - \lambda_\omega^N}{\Delta t} + \left[\frac{\partial \mathbf{R}^{N-1}}{\partial \mathbf{Q}^{N-1}} \right]^T \lambda_\omega^{N-1} = -\frac{\partial I^{N-1}}{\partial \mathbf{Q}^{N-1}} \quad \text{semiterminal condition, } (n = N - 1) \quad (33)$$

$$\frac{3\lambda_\omega^n - 4\lambda_\omega^{n+1} + \lambda_\omega^{n+2}}{2\Delta t} + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n} \right]^T \lambda_\omega^n = -\frac{\partial I^n}{\partial \mathbf{Q}^n} \quad \text{intermediate states, } (n = 2, \dots, N - 2) \quad (34)$$

$$\frac{3\lambda_\omega^1 - 4\lambda_\omega^2 + \lambda_\omega^3}{2\Delta t} = -\frac{\partial I^1}{\partial \mathbf{Q}^1} \quad \text{initial condition, } (n = 1) \quad (35)$$

where λ_ω is the adjoint of the vorticity field. Since the stream function equation [Eq. (30)] does not have an unsteady term, the adjoints of the stream function λ_ψ can be directly evaluated by

$$\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n} \right]^T \lambda_\psi^n = -\frac{\partial I^n}{\partial \mathbf{Q}^n} \quad \text{for } n = N, \dots, 1 \quad (36)$$

As discussed before, the preceding equations are solved backward in time to obtain the adjoint solutions. Finally, the global sensitivity of the augmented Lagrangian will be evaluated using Eq. (23).

The boundary conditions for the case of flow inside the cavity are straightforward, and the reader is referred to Ref. [42] for more details. However, extra care must be given to the treatment of the boundary conditions in the case of a cylinder. The boundaries for the cylinder in crossflow involve the freestream 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 freestream velocity at the far field such that $(u_\infty, v_\infty) = (U_\infty, 0.0)$. Assuming a unit velocity at the far field (i.e., $U_\infty = 1.0$), and based on the definition of the stream function, the flow variables at the far-field boundary are defined as

$$\omega_\infty = \omega(\eta = 1.0) = 0.0 \quad \psi_\infty = \psi(\eta = 1.0) = y \quad (37)$$

Furthermore, the no-slip wall condition requires a constant stream function value on the surface of the cylinder (i.e., $\psi(\eta = 0.0) = \text{constant}$), which is conveniently set to zero in many cases. Therefore,

$$\left. \frac{\partial \psi}{\partial \xi} \right|_{\eta=0.0} = 0$$

reducing Eq. (30) to

$$\omega(\eta = 0.0) + \beta \left. \frac{\partial^2 \psi}{\partial \eta^2} \right|_{\eta=0.0} + Q \left. \frac{\partial \psi}{\partial \eta} \right|_{\eta=0.0} = 0 \quad (38)$$

The preceding 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 a generalized coordinate system as

$$u_\theta = -\frac{\partial \psi}{\partial r} = -\left[\frac{\partial \psi}{\partial \xi} \frac{\partial \xi}{\partial r} + \frac{\partial \psi}{\partial \eta} \frac{\partial \eta}{\partial r} \right] \quad (39)$$

The first term on the right-hand side of the preceding equation must vanish due to a constant stream function value on a solid wall, thus simplifying it as

$$u_\theta = -\left[\frac{\partial \psi}{\partial \eta} \left(\frac{\partial \eta}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial \eta}{\partial y} \frac{\partial y}{\partial r} \right) \right] \rightarrow \frac{\partial \psi}{\partial \eta} = -\frac{u_\theta}{\eta_x \cos(\theta) + \eta_y \sin(\theta)} \quad (40)$$

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

$$\begin{aligned} u_{\text{rotational}} &= \Omega && \text{for the steady rotation case} \\ u_{\text{rotational}} &= \Omega \sin(2\pi \cdot f \cdot t) && \text{for the periodic rotation case} \end{aligned}$$

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

$$St = \frac{2r \cdot f}{U_\infty} \quad (41)$$

for a cylinder with a unit diameter (i.e., $r = 0.5$) subject to a unit freestream velocity $U_\infty = 1.0$.

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 crossflow 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

$$U_{\text{lid}}(t) = \frac{1}{\sin^{-1}(K)} \sin^{-1}(K \sin(2\pi f t)) \quad (42)$$

where K is the excitation parameter and f is the excitation frequency, taken to be $f = 1$ in this work. The excitation parameter is defined between $0 \leq K \leq 1$. For smaller values of the excitation parameter (i.e., $K \rightarrow 0$), the lid velocity will have a sinusoidal form, while for $K \rightarrow 1$, a triangular waveform with sharp crests is achieved. The primal governing equations are solved for the lid-driven cavity flow at a Reynolds number of $Re = 100$ on a rectangular grid with 101×101 equally spaced nodes and $\Delta t = 0.05$. The primal solutions of the CFD solver for a single period $T = 1/f = 1.0$ are presented in terms of the vorticity fields at $t = 0.25, 0.5, 0.75$, and 1.0 and are shown in Fig. 5.

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

$$\bar{I} = \frac{1}{T} \int_{t=0}^{t=T} \frac{1}{2} (\omega(t) - \omega_{\text{target}}(t))^2 dt = \frac{1}{T} \sum_{n=1}^N \frac{1}{2} (\omega^n - \omega_{\text{target}}^n)^2 \Delta t \quad (43)$$

where the vorticity ω and its target value ω_{target} are both defined at the midpoint of the lid boundary. The target time history of the unsteady vorticity response is taken to be the one obtained using $K = 0.9999$. Also, the initial value of the excitation parameter is taken to be $K = 0.8$, which corresponds to a semisinusoidal transient response.

First, we consider the adjoint solutions at four time instances during the interval of interest. These are taken to be at $t = 0.25, 0.5, 0.75$, and 1.0 , which correspond to the primal solutions presented earlier in Fig. 5. The adjoint solutions at these time instances are presented in Fig. 6. Since the objective function is dependent on the vorticity solution at the midpoint on the lid boundary, the adjoint solution field is also focused around this point. Additionally, the flow reversal phenomenon, which is typically seen in the adjoint solution, causes the sensitivities to backpropagate into the computational domain from the source of the objective function.

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. 7. As can be seen, the convergence of the 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 superlinear. On the other hand, the two cases with the local-in-time approach both have a lower convergence

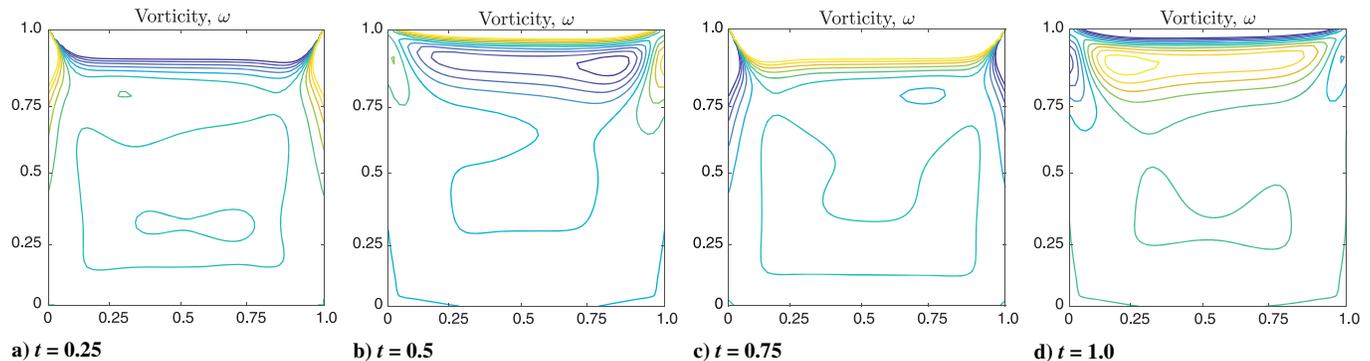


Fig. 5 Primal solutions of the vorticity field at four time steps during one period of excitation, $T = 1.0$.

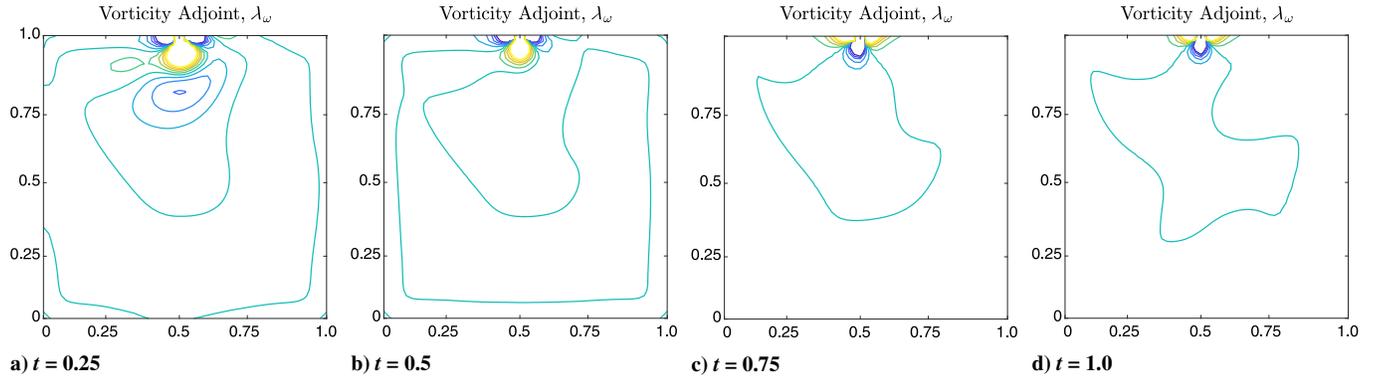


Fig. 6 Adjoint solutions of the vorticity field at four time steps during one period of excitation, $T = 1.0$.

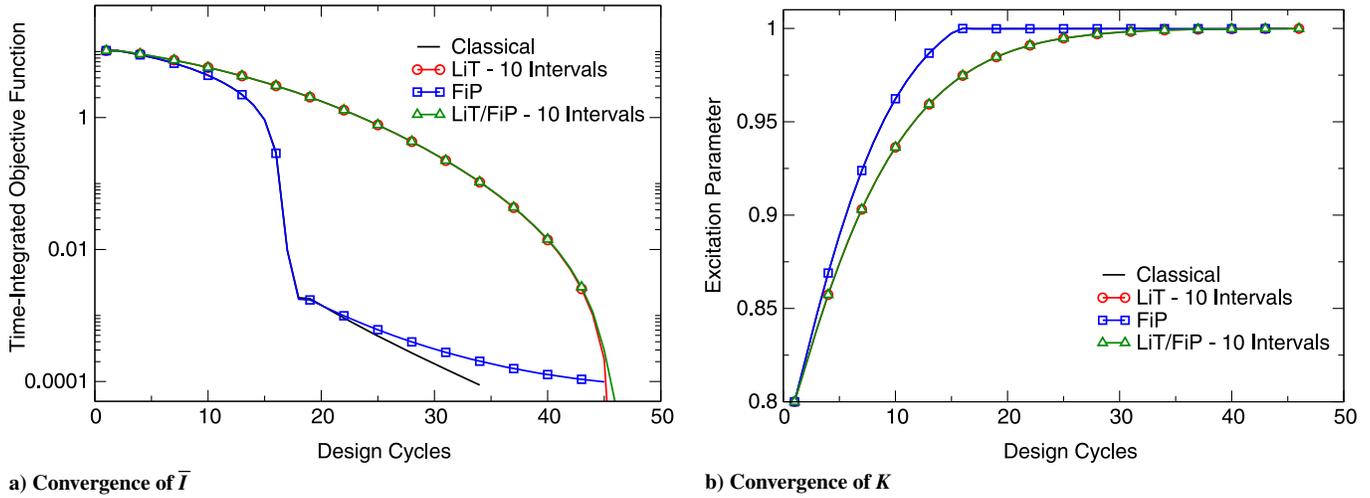


Fig. 7 Convergence of a) the time-integrated objective function and b) the excitation parameter for the unsteady design optimization problem using four different approaches.

rate as they slowly drive 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.

Furthermore, the computational cost associated with each method can be inferred from the convergence plot shown in Fig. 8 as a function of the wall-clock time. As can be seen, the fixed-point iteration approach results in the fastest convergence of the design

variable to its optimal solution, followed by the proposed hybrid approach that also uses a local-in-time technique with 10 subintervals. Additionally, normalized CPU times and memory footprints are presented in Table 1 for the different approaches used in the inverse design of the cavity flow. These results show the robustness of the proposed technique for one-shot unsteady design. It is apparent that the approach results in significant performance gains in terms of both computational cost and memory efficiency.

It must be noted that for the LiT and LiT/FiP approaches, cases with 2, 4, and 10 subintervals are considered, while only the results with 10 subintervals were presented in Fig. 7. To study the effects of the number of subintervals on the performance of the unsteady one-shot design optimization approach, cases with different numbers of subintervals are compared next. These results are presented in Figs. 9 and 10 in terms of the objective function and excitation parameter convergence for the LiT and LiT/FiP approaches where the local-in-time technique is used. Interestingly, for both the LiT and LiT/FiP

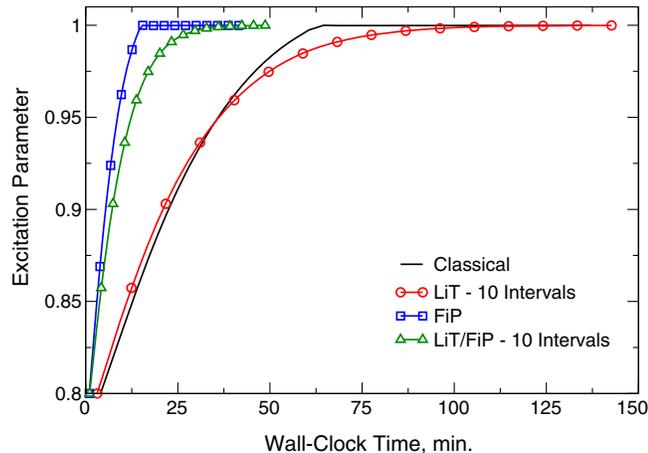
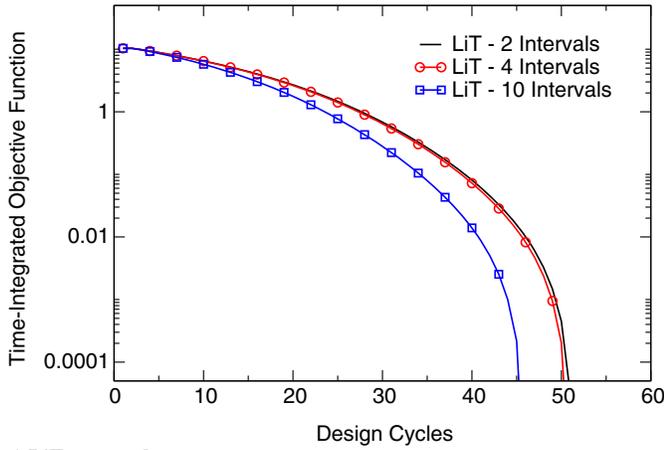


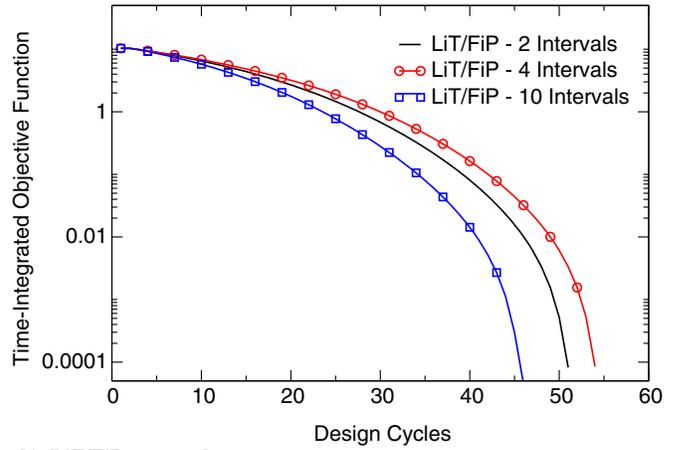
Fig. 8 Convergence of the excitation parameter (design variable) for the unsteady design optimization problem using four different approaches in terms of wall-clock time.

Table 1 Normalized CPU times and memory footprints for different unsteady one-shot approaches used for the inverse design of the cavity flow

| One-shot approach | Normalized time | Normalized memory footprint | Reduction, % |
|-----------------------|-----------------|-----------------------------|--------------|
| Classical | 1.0 | 1.0 | — |
| LiT, 10 intervals | 1.04 | 0.1 | 90 |
| FiP | 0.317 | 1.0 | — |
| LiT/FiP, 10 intervals | 0.355 | 0.1 | 90 |

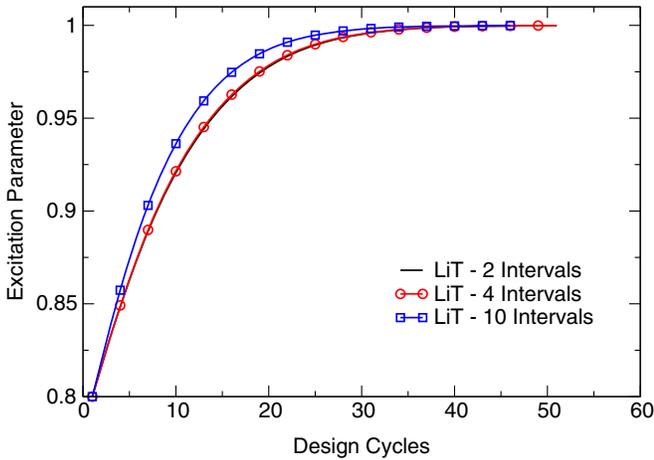


a) LiT approach

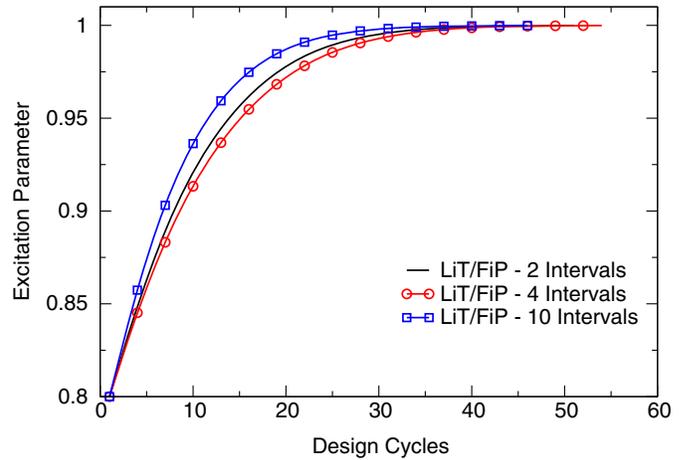


b) LiT/FiP approach

Fig. 9 Effects of the number of subintervals on the convergence of the time-integrated objective function for the unsteady design optimization problem using the a) LiT and b) LiT/FiP approaches.



(a) LiT approach



(b) LiT/FiP approach

Fig. 10 Effects of the number of subintervals on the convergence of the excitation parameter for the unsteady design optimization problem using the a) LiT and b) LiT/FiP approaches.

approaches, cases with 10 subintervals lead to faster convergence while also providing an almost 90% reduction in memory footprint.

Finally, the time history of the vorticity at the midpoint 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. 11. It is seen that the inverse design optimization leads to a vorticity distribution that perfectly matches the target 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. 11.

More importantly, the FiP approach provides a more than 40% reduction 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 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 time-accurate design optimization.

B. Vortex Suppression and Mean Drag Reduction for Cylinder in Crossflow

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 $Re_{crit} > 47$ in a so-called Kármán vortex street [43].

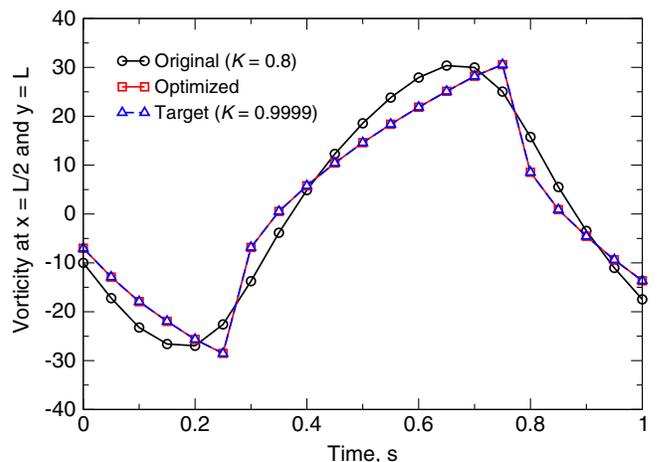


Fig. 11 Comparison of unsteady vorticity solutions for the original ($K = 0.8$), optimized, and target ($K = 0.9999$) designs of the cavity flow with unsteady lid velocity.

With a deeper understanding of the flow features in the wake of the cylinder, focus has been shifted toward controlling the vortex shedding by rotating the cylinder. It has been understood that certain conditions lead to vortex suppression [44,45]. The experimental work of Tokumaru and Dimotakis [46] proved that finding an optimal rotational velocity for the cylinder in crossflow may result in significant mean drag reductions. This effect has been also studied numerically by Kang et al. and He et al. [47,48] for various Reynolds numbers. Additionally, in the area of unsteady adjoint-based design, several studies have focused on vortex suppression or mean drag minimization for a circular cylinder via constant or time-periodic rotation of the cylinder [19,49,50].

In this work, the use of the proposed unsteady one-shot approach is investigated to identify an optimal rotational velocity that can either lead to vortex suppression in the wake or a reduction in the mean drag coefficient. Here, flow past the circular cylinder with a unit diameter at $Re = 100$ is considered. For the case with steady rotation, the physical domain is discretized by an O-type grid with 177×81 nodes in the circumferential ξ and normal η directions, respectively. The computational domain is extended for 100 diameters, and the grid in the vicinity of the cylinder is shown in Fig. 12a.

1. Vortex Suppression via Steady Rotation

The first optimal control problem considers a steady rotation. Here, the cylinder rotational velocity is defined as $u_{\text{rotational}} = \Omega$, which is chosen as the design variable. As mentioned earlier, the goal of this test case is to study vortex suppression via steady rotation. Here, we follow an approach similar to the one proposed by Homescu et al. [49] in which a target stream function distribution is considered in the framework of a flow-tracking procedure. For this reason, the flowfield at $Re = 2$ with a steadily rotating cylinder at $\Omega = 2$ is chosen [49]. With that, the objective function for the flow-tracking approach can be written as

$$\bar{I} = \frac{1}{T} \int_0^T \frac{1}{2} \int_{\mathcal{V}} [\psi - \psi^{\text{target}}]^2 d\mathcal{V} dt \quad (44)$$

where \mathcal{V} refers to the entire computational domain and ψ^{target} is the stream function field for the target solution (i.e., $Re = 2$ 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. [49], this can lead to an ill-posed problem in which the optimal rotational speed would continually increase. Therefore, a regularization function, similar to that proposed by Homescu et al. [49], will be used to resolve the ill-posedness. The regularization function limits the rotational velocity on the surface of the cylinder by augmenting the objective function such that

$$\bar{I} = \frac{1}{T} \int_0^T \frac{1}{2} \int_{\mathcal{V}} [\psi - \psi^{\text{target}}]^2 d\mathcal{V} dt + R_F \frac{1}{T} \int_0^T \frac{1}{2} \oint_{\mathcal{S}} [u_{\text{rotational}}]^2 d\mathcal{S} dt \quad (45)$$

where \mathcal{S} refers to the cylinder surface boundary and R_F is a constant 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. [49] have shown that a trial and error technique can be used to determine the regularization factor for various Reynolds numbers. A similar approach has been adopted here leading to $R_F = 1500$ for the $Re = 100$ case presented in this section.

First, the primal flowfield with $\Omega = 1.0$ is solved for 150 s 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 s time interval is chosen as the design window. However, before presenting the design optimization results, it is important to make sure that the primal solutions are spatially and temporally converged by conducting a grid and time-step convergence study. In this regard, the grid resolution is doubled in both the radial and circumferential directions to achieve a finer (353×161) grid. Also, the time step is halved to study the temporal accuracy. The resulting lift coefficient time histories using the doubled spatial and temporal resolutions are compared to the nominal grid resolution and time step during the design interval. These results are shown in Fig. 13, which clearly exhibits the convergence of the primal solutions in both space and time.

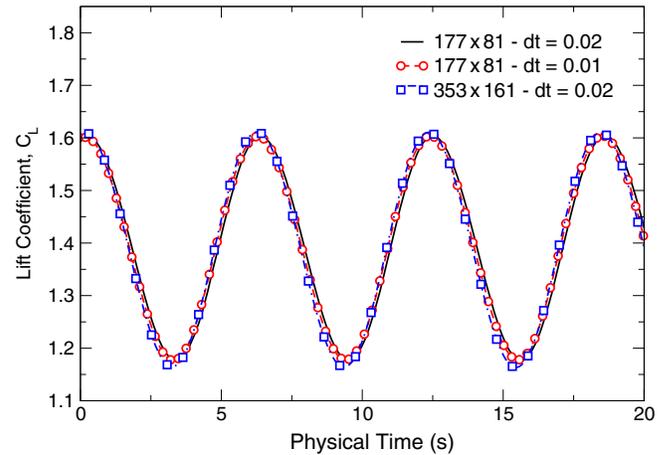


Fig. 13 Spatial and temporal convergence analysis based on the unsteady lift history in interval [0, 20] (cylinder in crossflow with steady rotation).

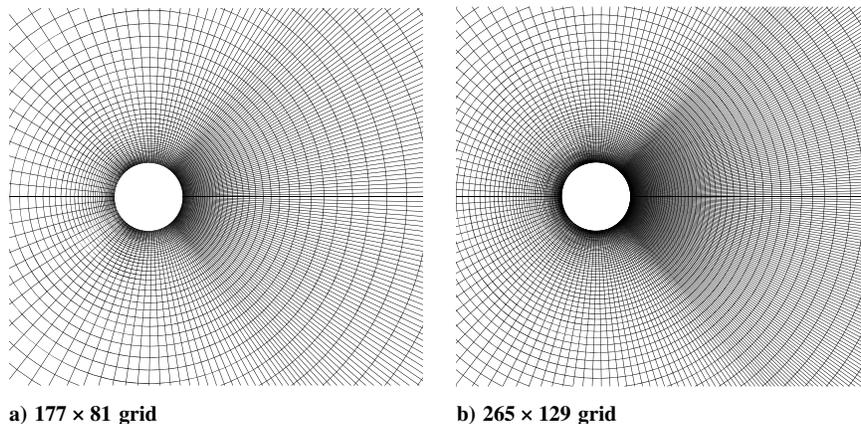


Fig. 12 Computational grids with node clustering in the wake region used for the flow past the circular cylinder with a) steady and b) periodic rotation settings.

In addition to the qualitative convergence study presented herein, a quantitative study is also performed in terms of the mean lift coefficients as well as the Strouhal numbers. These results are presented in Table 2 and once again prove that the primal solutions are converged both spatially and temporally. It must be noted that this convergence study is crucial for any unsteady simulation. Additionally, the accuracy of the discrete adjoint solutions is directly related to the accuracy of the primal flow solutions, thus making the convergence study presented herein even more crucial for the validity of the unsteady design solutions sought in this work.

Similar to the lid-driven cavity optimization that was presented earlier, the efficacy of the proposed hybrid technique (LiT/FiP) is investigated. For the local-in-time approach, varying numbers of subintervals are tested. Here, for the design window of $[0, 20]$, the number of subintervals is taken to be $M = 2, 4, 10,$ and 25 for both the LiT and LiT/FiP approaches. With the choice of $R_F = 1500$, the

Table 2 Spatial and temporal convergence study for the cylinder with steady rotation at $Re = 100$

| Case | Mean C_L | % difference | Sr | % difference |
|------------------------------------|------------|--------------|----------|--------------|
| $177 \times 81 - \Delta t = 0.02$ | 1.38935 | — | 0.165046 | — |
| $177 \times 81 - \Delta t = 0.01$ | 1.38989 | 0.0388 | 0.165026 | 0.0121 |
| $353 \times 161 - \Delta t = 0.02$ | 1.38626 | 0.2224 | 0.166121 | 0.6513 |

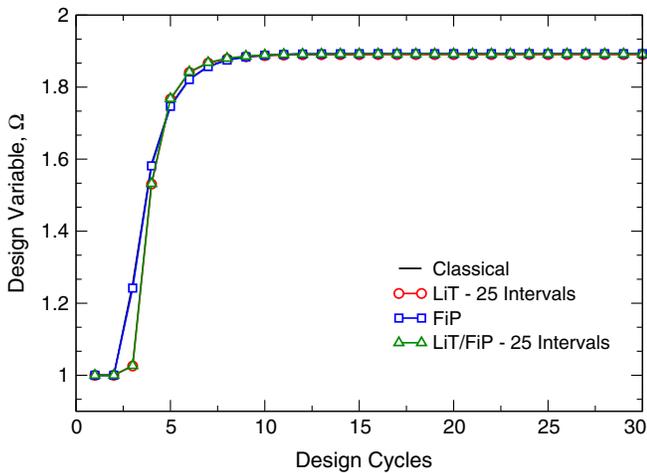


Fig. 14 Convergence of the quadratic programming problem for the optimal steady rotation rate of the cylinder in crossflow at $Re = 100$.

sequential least-squares quadratic programming method converges to an optimal rotational velocity amplitude of $\Omega_{opt} = 1.8932$. The convergence histories of the optimizer using the classical approach (GiT), the LiT method with 25 intervals, the FiP approach, and the proposed LiT/FiP technique also with 25 subintervals are presented in Fig. 14. As can be seen, all four approaches converge to the same optimal solution, while the cases using a local-in-time approach exhibit a slightly slower convergence during the early design cycles. This behavior is associated with the fact that the LiT approach is prone to slight errors in early stages, mainly because of the discontinuities present in the backward-in-time procedure. Simply, the terminal conditions for these equations rely on adjoint solutions from the next subinterval that are yet to be determined. As such, an error is introduced, but that error gets smaller with the number of design cycles.

Next, the effect of the number of subintervals on the convergence behavior of the unsteady one-shot techniques is studied. Here, the LiT and LiT/FiP approaches with various numbers of subintervals 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 subintervals can have a slight degradation of the convergence behavior at the early stages.

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 to note that in all cases, the gradients of the time-averaged objective function are reduced significantly. Additionally, the LiT

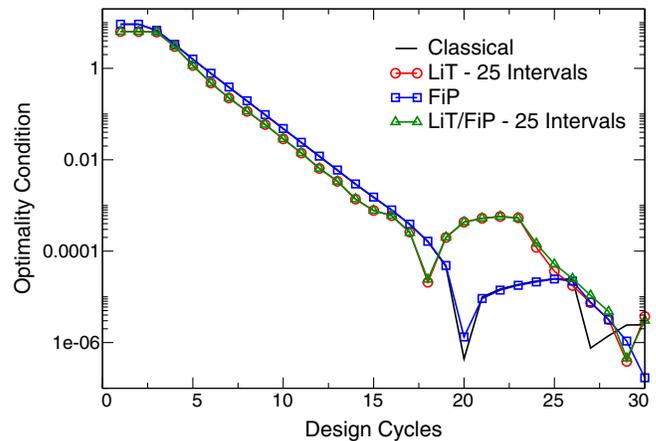
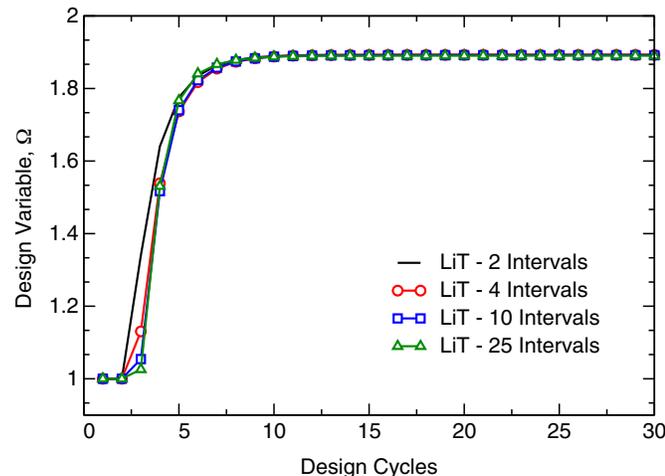
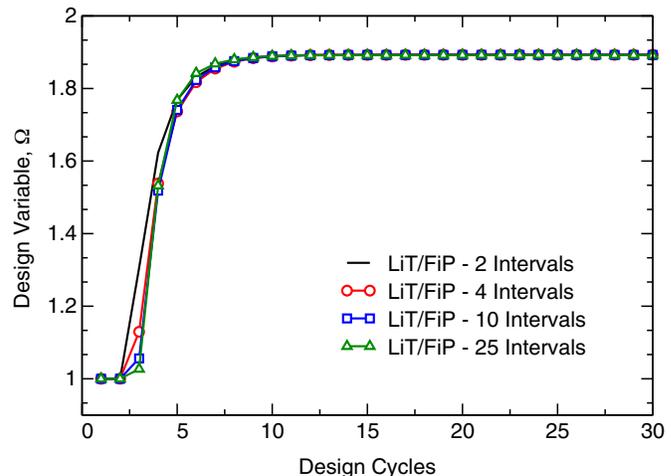


Fig. 16 Optimality condition for the quadratic programming used in determining the optimal steady rotation rate of the cylinder in crossflow at $Re = 100$.



a) LiT approach



b) LiT/FiP approach

Fig. 15 Effects of the number of subintervals on the unsteady design optimization using the a) LiT and b) LiT/FiP approaches.

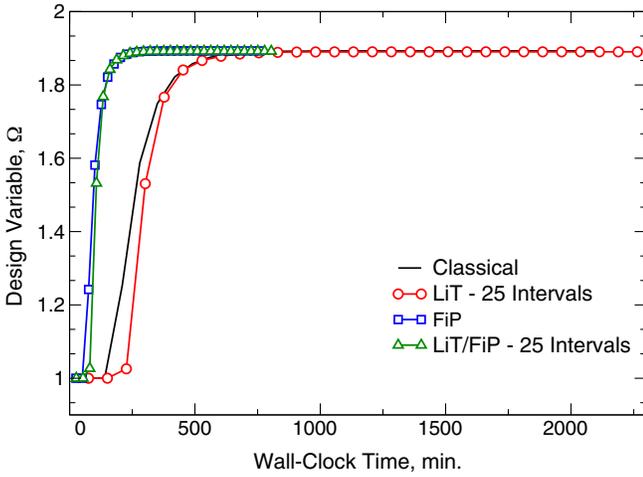


Fig. 17 CPU times for different unsteady one-shot techniques used in this work for the optimal control of the vortex shedding of a circular cylinder at $Re = 100$ via a steady rotation.

Table 3 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 intervals | 1.0 | LiT/FiP, 2 intervals | 0.33 |
| LiT, 4 intervals | 1.04 | LiT/FiP, 4 intervals | 0.35 |
| LiT, 10 intervals | 1.05 | LiT/FiP, 10 intervals | 0.36 |
| LiT, 25 intervals | 1.08 | LiT/FiP, 25 intervals | 0.38 |

Table 4 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 and LiT/FiP, 2 intervals | 0.5 | 50 |
| LiT and LiT/FiP, 4 intervals | 0.25 | 75 |
| LiT and LiT/FiP, 10 intervals | 0.1 | 90 |
| LiT and LiT/FiP, 25 intervals | 0.04 | 96 |

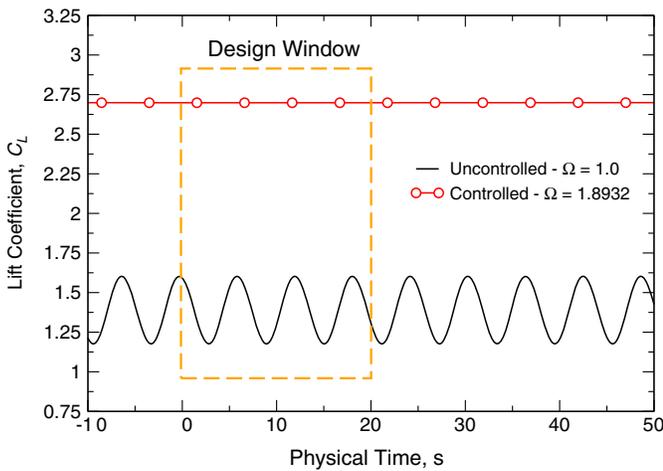
and LiT/FiP techniques, which rely on a local-in-time approach, exhibit a different behavior at the late stages due to the possible undershooting and overshooting of the optimizer close to the optimal solution. In fact, Rumpfkeil and Zingg [19] have shown that the design space for this control problem has various local optima that are close to each other, leading to such behavior.

As discussed earlier in this paper, the 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 LiT approach is shown to provide consistent reductions in memory footprint as the number of subintervals is increased. The results in terms of computational and memory savings are presented in Tables 3 and 4. As can be seen, the proposed LiT/FiP approach can provide up to a 65% reduction in the computational time as well as a 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.

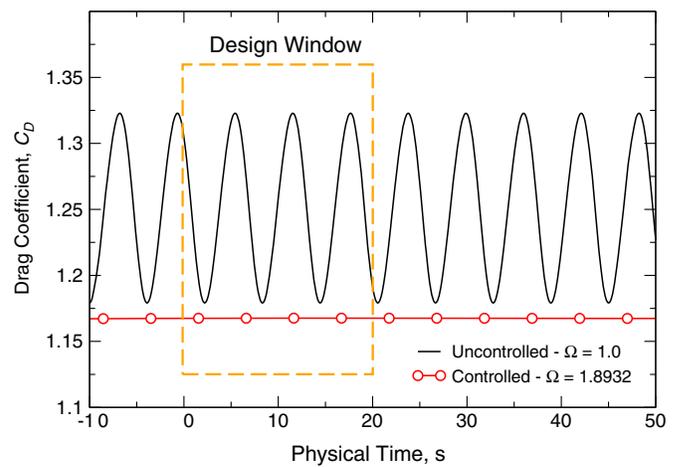
It must be noted that Homescu et al. [49] have used a similar design approach that has led to an optimal rotation amplitude of $\Omega = 1.84$. On the other hand, Kang et al. [47] have reported $\Omega = 1.9$ as the optimal solution, which they achieved by gradually increasing the rotational velocity amplitude until the drag variations are significantly reduced. As can be seen, the optimal solution obtained in this work, $\Omega_{\text{opt}} = 1.8932$, is in very good agreement with the values reported in the literature.

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 Fig. 18. As mentioned earlier, the unsteady optimization problem is hot started by using the periodic flow solution after 150 s. This means that the time interval of $[0, 20]$ used as the design window is in fact the time interval between $[150, 170]$ s. 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.

Finally, the vorticity fields and the streamlines around the cylinder with the original and optimal surface rotational velocities are shown in Fig. 19. It can be clearly seen that the vortex street is completely suppressed in the controlled case in which the cylinder is rotated at a steady rate of $\Omega_{\text{opt}} = 1.8932$, which is found to be the optimal value. It must be noted that this value is a function of the regularization parameter, which limits the maximum value of the rotational speed.



a) Lift coefficient, C_L



b) Drag coefficient, C_D

Fig. 18 Time histories of lift and drag coefficients for original (uncontrolled with $\Omega_{\text{orig}} = 1.0$) and optimal (controlled with $\Omega_{\text{opt}} = 1.8932$) cases of cylinder in crossflow at $Re = 100$ with steady rotation.

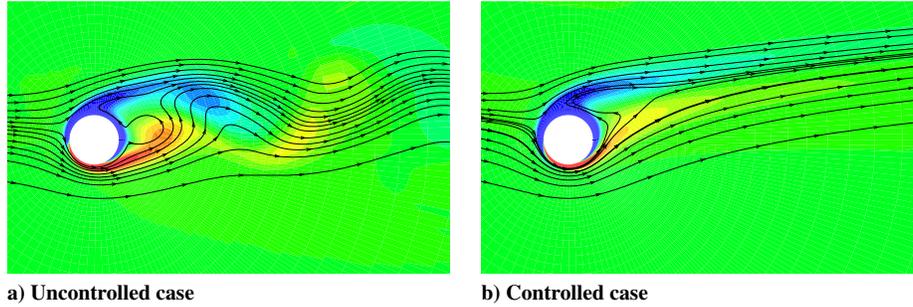


Fig. 19 Vorticity contour field and the streamlines for original (uncontrolled with $\Omega_{\text{orig}} = 1.0$) and optimal (controlled with $\Omega_{\text{opt}} = 1.8932$) cases of cylinder in crossflow at $Re = 100$ with steady rotation.

2. Mean Drag Reduction via Time-Periodic Rotation

Having presented the results of the unsteady design optimization pertaining to the vortex suppression of a steadily rotating cylinder in crossflow, we now shift our attention to the case in which 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 [i.e., $u_{\text{rotational}} = \Omega \sin(2\pi \cdot f \cdot t)$], 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 flowfield used for hot starting the design process corresponds to the natural vortex shedding of the stationary cylinder at $Re = 100$. For the present test case, a finer grid with 265×129 nodes, shown in Fig. 12b, is used 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 $[0, 20]$ s, and a physical time step of $\Delta t = 0.01$ is used.

Unlike the previous test case, in which 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

$$\bar{I} = \frac{1}{T} \sum_{n=1}^N C_D(x^n, Q^n(x)) \Delta t \quad (46)$$

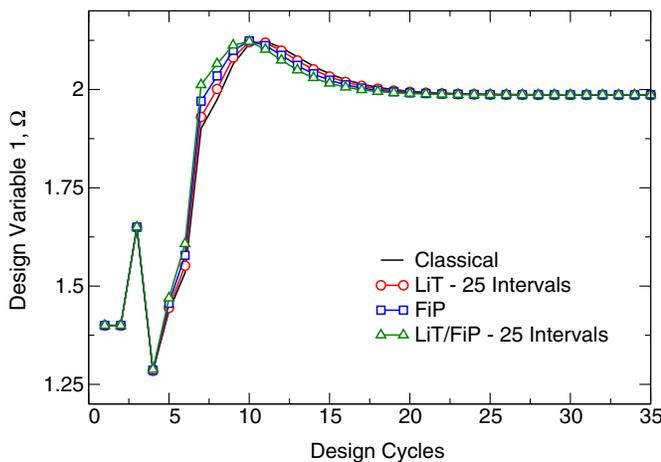
where C_D is the drag coefficient calculated based on the instantaneous flow solution and the design variables (i.e., oscillation parameters Ω and f). It must be noted that a similar approach was adopted by other researchers to reduce the mean drag for an oscillating cylinder at various Reynolds numbers [19, 47, 48, 50].

In this work, the TAPENADE automatic differentiation toolbox [40] is used to calculate the sensitivities of the drag coefficient with

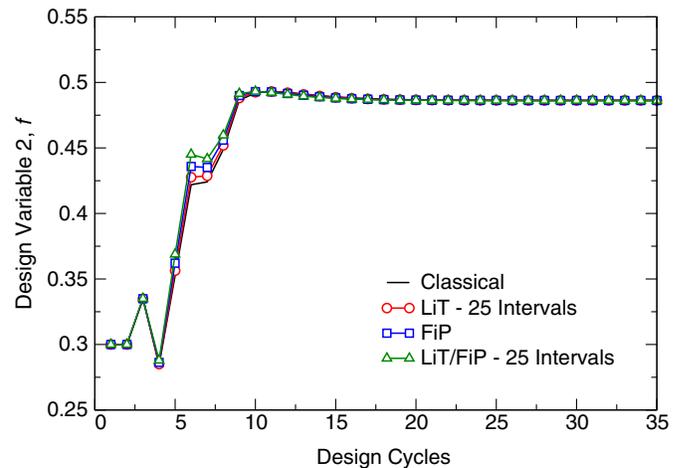
respect to the flow solution and the design variables. Therefore, the automatically differentiated functions are called during backward-in-time adjoint solutions to obtain $\partial C_D^n / \partial Q^n$ and $\partial C_D^n / \partial x^n$ terms based on the current flow solutions and design variables at time level n . As discussed earlier, the sensitivities of the objective function with respect to the flow variables (i.e., $\partial I / \partial Q$) are used on the right-hand side of the adjoint equations [see Eqs. (32-35)] 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., $\partial I / \partial x$) are included in the total derivative accumulation according to Eq. (23).

Once again, the primal flow solver is run for 100 s 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 subintervals are used for the LiT and LiT/FiP approaches, which would require 80 time steps during each subinterval and can provide a more than 90% reduction in the memory footprint.

First, the convergence behaviors for both design variables (i.e., Ω and f) are presented in Fig. 20 using the classical (GiT), LiT, and FiP techniques, 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 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 subinterval 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 $f_{\text{opt}} = 0.486$.

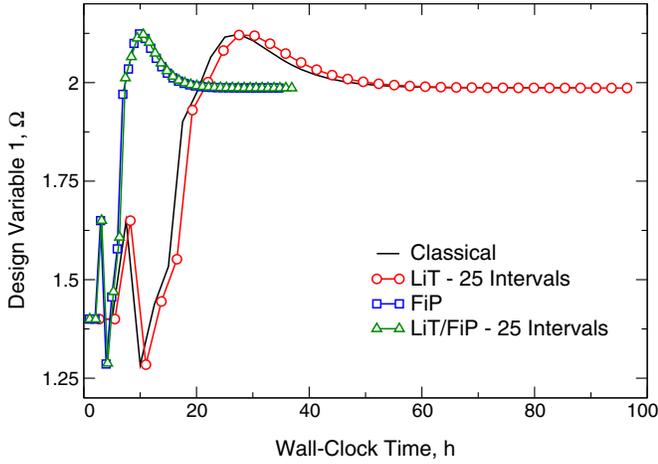


a) Ω convergence

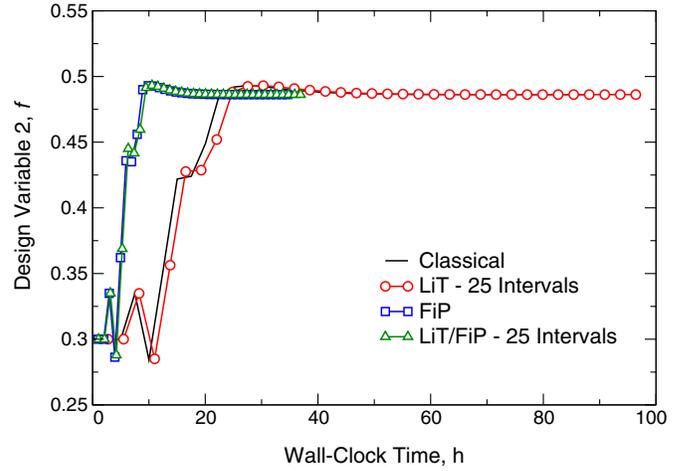


b) f convergence

Fig. 20 Convergence of the quadratic programming problem for rotational velocity amplitude Ω and forcing frequency f in the optimal periodic rotation of the cylinder in crossflow at $Re = 100$.



a) Ω convergence



b) f convergence

Fig. 21 Convergence histories of the two design variables (Ω and f) in terms of wall-clock times for the optimal periodic rotation of the cylinder in crossflow at $Re = 100$.

On the other hand, similar computational and memory savings are achieved with the FiP, LiT, and the proposed LiT/FiP approaches, as was shown in the previous test case. The convergence histories in terms of the CPU times as well as normalized times and memory footprints are shown in Fig. 21 and Table 5. 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.

Next, the optimality conditions for both design variables are presented in Fig. 22 using different unsteady one-shot techniques.

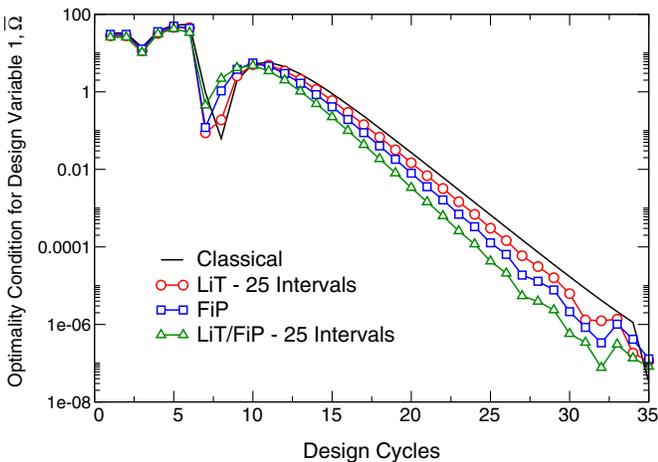
Table 5 Normalized CPU times and memory footprints for different unsteady one-shot approaches used for the mean drag reduction of the oscillating cylinder in crossflow at $Re = 100$

| One-shot approach | Normalized time | Normalized memory footprint | Reduction, % |
|-----------------------|-----------------|-----------------------------|--------------|
| Classical | 1.0 | 1.0 | — |
| LiT, 10 intervals | 1.10 | 0.1 | 90 |
| FiP | 0.395 | 1.0 | — |
| LiT/FiP, 10 intervals | 0.421 | 0.1 | 90 |

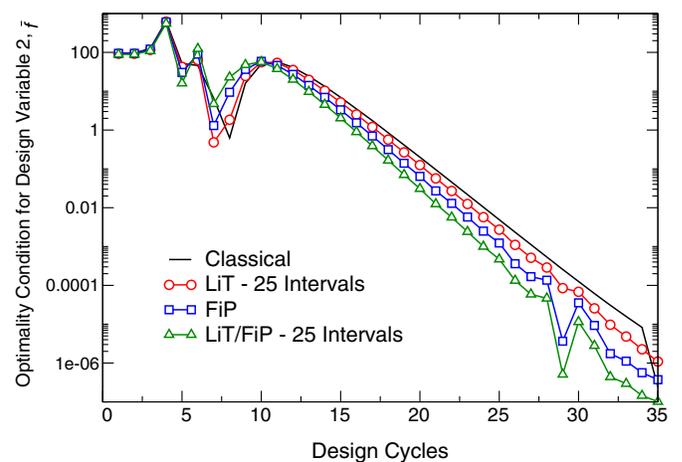
Once again, it is shown that the proposed hybrid LiT/FiP approach does not negatively affect the optimality or convergence behavior of the quadratic programming. Additionally, in all cases studied here, the convergence of the optimization problem to the optimal solutions is guaranteed, although the rate 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 and 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 $\Omega_{opt} = 1.986$ and $f_{opt} = 0.486$. 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.

It is worth noting that the optimal values obtained in this work are different from the optimal solutions reported by Homescu et al. [49] ($\Omega_{Homescu} \approx 6.5$ and $f_{Homescu} = 1.13$) and He et al. [48] ($\Omega_{He} \approx 6.0$ and $f_{He} = 0.75$). Interestingly, Rumpfkeil and Zingg [19] and Mahmood et al. [50] also report very different optimal values for the rotational velocity amplitude and the forcing frequency. The former work also presents 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 [19]. Therefore, it can be understood that the drag minimization of the oscillatory cylinder is a multimodal

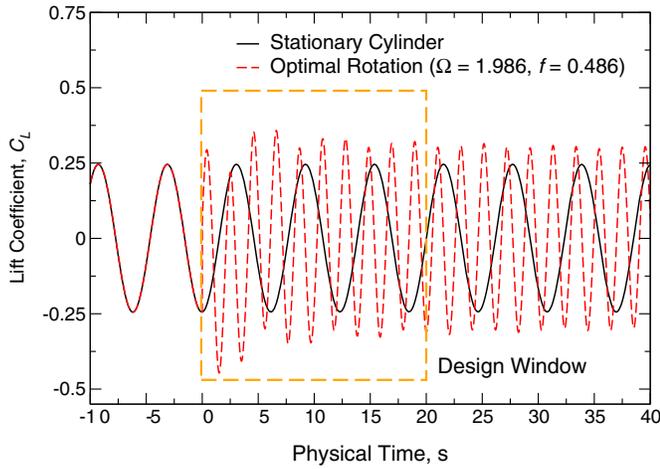


a) $\frac{\partial I}{\partial \Omega}$ convergence

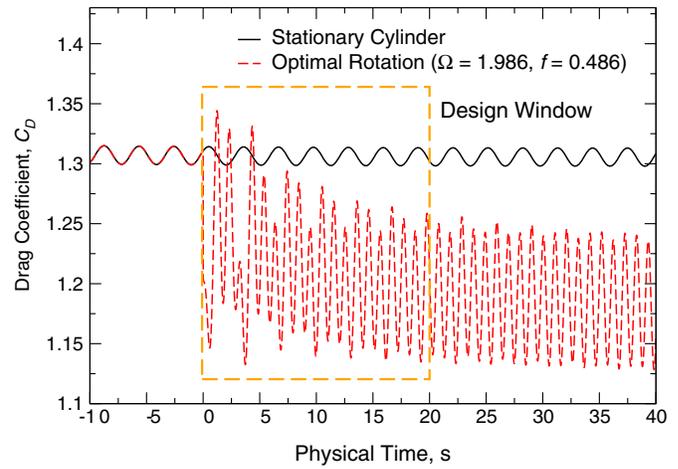


b) $\frac{\partial I}{\partial f}$ convergence

Fig. 22 Optimality conditions for the quadratic programming used in determining optimal rotational velocity amplitude and forcing frequency of the oscillating cylinder in crossflow at $Re = 100$.



a) Lift coefficient, C_L



b) Drag coefficient, C_D

Fig. 23 Time histories of the lift and drag coefficients for original (stationary) and optimal (oscillatory with $\Omega_{opt} = 1.986$, $f_{opt} = 0.486$) cases of cylinder in crossflow at $Re = 100$.

optimization problem in which, depending on the initial condition, it is possible to converge to different local optima. It would be interesting to study the multimodality 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.

Next, the lift and drag force time histories for the original and optimal designs are shown in Fig. 23. 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 [19, 48, 49]. The detailed comparisons between the lift and drag variations as well as the reduction in the mean drag coefficient are presented in Table 6.

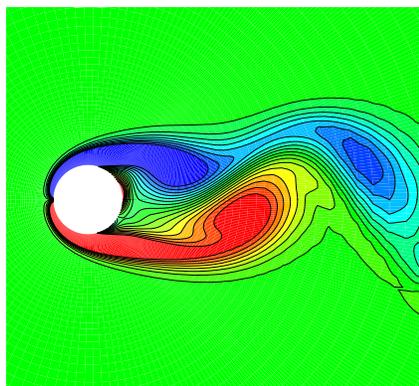
Table 6 Comparison of the lift and drag variations and mean drag reduction between the original (stationary) and optimal oscillatory rotation of the cylinder at $Re = 100$

| 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% |

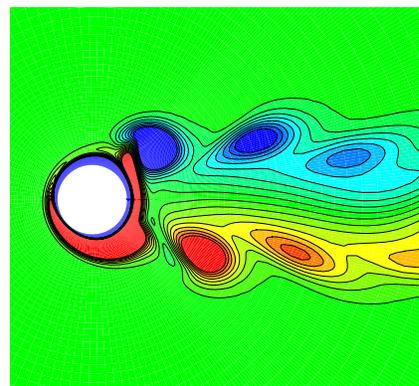
Finally, the effect of the optimal periodic rotation of the cylinder on the vortex dynamics can be seen in the vorticity contour plots shown in Fig. 24. As can be seen here, unlike the steady rotation design case presented earlier, a complete suppression of the vortex shedding has not been achieved. 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 quasimagnetized [48]. 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 of 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 subintervals that 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 used that can substantially improve the computational efficiency in each design cycle. It must be noted that the proposed hybrid technique was developed as a result of successive improvements of the numerical procedures starting from the classical approach. By using the fixed-point and local-in-time techniques separately in the framework of an unsteady design optimization problem, it is proven possible to achieve considerable savings in



a) Uncontrolled (stationary) case



b) Controlled (optimal oscillation) case

Fig. 24 Vorticity contour field for original (uncontrolled with natural vortex shedding from the stationary cylinder) and optimal (controlled with $\Omega_{opt} = 1.986$, $f_{opt} = 0.486$) cases of cylinder in crossflow at $Re = 100$.

terms of computational cost as well as memory footprint. Therefore, the ultimately proposed method was developed by combining the two individual techniques into a hybrid approach that can provide both computational and memory efficiencies. During the development of this method, it was found that hot starting the adjoint solutions in the LiT approach did not have significant effects on the accuracy of the adjoint calculations. Therefore, the terminal conditions for the adjoint solutions at the end of each subinterval can simply rely on an adjoint flow initialization due to the use of a direct linear solution method to obtain the λ vector at each time step. The proposed hybrid technique, called LiT/FiP, is used to tackle several unsteady optimization problems involving the lid-driven cavity flow and a circular cylinder in crossflow. Unsteady optimization results using the LiT/FiP technique are compared to those obtained from the classical approach as well as cases with only the FiP and LiT techniques being used. It has been shown that the proposed technique can provide up to a 65% reduction in the computational time by using the fixed-point-iteration technique. At the same time, the use of the local-in-time technique with varying numbers of subintervals has shown to provide a 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.

Acknowledgment

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] Christianson, B., "Reverse Accumulation and Implicit Functions," *Optimization Methods and Software*, Vol. 9, No. 4, 1998, pp. 307–322.
- [2] Giannakoglou, K. C., and Papadimitriou, D. I., "Adjoint Methods for Shape Optimization," *Optimization and Computational Fluid Dynamics*, Springer, Berlin, 2008, pp. 79–108.
- [3] Giles, M. B., and Pierce, N. A., "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion*, Vol. 65, Nos. 3–4, 2000, pp. 393–415.
- [4] Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S., *Optimization with PDE Constraints*, Vol. 23, Springer Science and Business Media, Berlin, 2008.
- [5] Huang, H., and Ekici, K., "A Discrete Adjoint Harmonic Balance Method for Turbomachinery Shape Optimization," *Aerospace Science and Technology*, Vol. 39, Dec. 2014, pp. 481–490.
- [6] 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, Aug. 2019, pp. 159–174.
- [7] 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, Berlin, 2003, pp. 3–13.
- [8] Özkaya, E., "One-Shot Methods for Aerodynamic Shape Optimization," Ph.D. Thesis, RWTH Aachen Univ., Aachen, Germany, 2014.
- [9] Thress, J. F., Kaminsky, A. L., Djeddi, R., and Ekici, K., "One-Shot Design Optimization Based on the Adjoint Harmonic Balance Technique," AIAA Paper 2020-3128, 2020.
- [10] Ta'asan, S., "Pseudo-Time Methods for Constrained Optimization Problems Governed by PDE," Inst. for Computer Applications in Science and Engineering, Tech. Rept. ADA296242, Hampton, VA, May 1995.
- [11] 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.
- [12] Hazra, S. B., "Aerodynamic Shape Optimization Using Simultaneous Pseudo-Time-Stepping," *Large-Scale PDE-Constrained Optimization in Applications*, Springer, Berlin, 2010, pp. 81–104.
- [13] Duta, M., Giles, M., and Campobasso, M., "The Harmonic Adjoint Approach to Unsteady Turbomachinery Design," *International Journal for Numerical Methods in Fluids*, Vol. 40, Nos. 3–4, 2002, pp. 323–332.
- [14] Thomas, J. P., Hall, K. C., and Dowell, E. H., "Discrete Adjoint Approach for Modeling Unsteady Aerodynamic Design Sensitivities," *AIAA Journal*, Vol. 43, No. 9, 2005, pp. 1931–1936.
- [15] 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.
- [16] Choi, S., Potsdam, M., Lee, K., Iaccarino, G., and Alonso, J., "Helicopter Rotor Design Using a Time-Spectral and Adjoint-Based Method," AIAA Paper 2008-5810, 2008.
- [17] Mohammadi, B., and Pironneau, O., *Applied Shape Optimization for Fluids*, Oxford Univ. Press, Oxford, 2010.
- [18] 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.
- [19] 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.
- [20] 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, Berlin, 2011, pp. 318–327.
- [21] Lee, B. J., and Liou, M.-S., "Unsteady Adjoint Approach for Design Optimization of Flapping Airfoils," *AIAA Journal*, Vol. 50, No. 11, 2012, pp. 2460–2475.
- [22] Ntanakas, G., and Meyer, M., "Towards Unsteady Adjoint Analysis for Turbomachinery Applications," *Proceedings of the 6th European Conference on Computational Fluid Dynamics-ECFD VI*, Barcelona, Spain, July 2014, pp. 5071–5081.
- [23] Economou, T. D., Palacios, F., and Alonso, J. J., "Unsteady Continuous Adjoint Approach for Aerodynamic Design on Dynamic Meshes," *AIAA Journal*, Vol. 53, No. 9, 2015, pp. 2437–2453.
- [24] Zhou, B. Y., Albring, T. A., Gauger, N. R., Economou, T. D., and Alonso, J. J., "An Efficient Unsteady Aerodynamic and Aeroacoustic Design Framework Using Discrete Adjoint," AIAA Paper 2016-3369, 2016.
- [25] Ma, C., Su, X., and Yuan, X., "An Efficient Unsteady Adjoint Optimization System for Multistage Turbomachinery," *Journal of Turbomachinery*, Vol. 139, No. 1, 2017, Paper 011003.
- [26] Rubino, A., Pini, M., Colonna, P., Albring, T., Nimmagadda, S., Economou, T., and Alonso, J., "Adjoint-Based Fluid Dynamic Design Optimization in Quasi-Periodic Unsteady Flow Problems Using a Harmonic Balance Method," *Journal of Computational Physics*, Vol. 372, Nov. 2018, pp. 220–235.
- [27] 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.
- [28] Krakos, J. A., "Unsteady Adjoint Analysis for Output Sensitivity and Mesh Adaptation," Ph.D. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, 2012.
- [29] 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*, Vol. 26, No. 1, 2000, pp. 19–45.
- [30] 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.
- [31] 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.
- [32] Hüchelheim, 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, Berlin, 2019, pp. 117–130.
- [33] Wang, Q., "Adjoint Sensitivity Analysis of Time Averaged Quantities for Unsteady Flows," *APS Division of Fluid Dynamics Meeting Abstracts*, Vol. 56, No. 18, Nov. 2011, pp. H19–001.
- [34] Schotthöfer, S., Zhou, B. Y., Albring, T. A., and Gauger, N. R., "Windowing Regularization Techniques for Unsteady Aerodynamic Shape Optimization," AIAA Paper 2020-3130, 2020.
- [35] 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.
- [36] Yamaleev, N., Diskin, B., and Nielsen, E., "Adjoint-Based Methodology for Time-Dependent Optimization," AIAA Paper 2008-5857, 2008.
- [37] Beran, P., Stanford, B., and Kurdi, M., "Sensitivity Analysis for Optimization of Dynamic Systems with Reduced Order Modeling," AIAA Paper 2010-1503, 2010.

- [38] 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, March 2016, pp. 12–22.
- [39] Nocedal, J., and Wright, S., *Numerical Optimization*, Springer Science & Business Media, Berlin, 2006.
- [40] Hascoët, L., "TAPENADE: A Tool for Automatic Differentiation of Programs," *Proceedings of 4th European Congress on Computational Methods, ECCOMAS*, Jyväskylä, Finland, July 2004, pp. 1–14.
- [41] Günther, S., "Simultaneous Optimization with Unsteady Partial Differential Equations," Ph.D. Thesis, RWTH Aachen Univ., Aachen, Germany, 2017.
- [42] 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.
- [43] 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*, 1st ed., Vol. 1911, Jan. 1912, pp. 509–517.
- [44] 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, Sept. 1989, pp. 579–627.
- [45] Henderson, R. D., "Nonlinear Dynamics and Pattern Formation in Turbulent Wake Transition," *Journal of Fluid Mechanics*, Vol. 352, Dec. 1997, pp. 65–112.
- [46] Tokumaru, P., and Dimotakis, P., "Rotary Oscillation Control of a Cylinder Wake," *Journal of Fluid Mechanics*, Vol. 224, March 1991, pp. 77–90.
- [47] 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.
- [48] 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.
- [49] Homesu, 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.
- [50] 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, pp. 4825–4843.

K. E. Willcox
Associate Editor