International Journal for Numerical Methods in Engineering

WILEY

RESEARCH ARTICLE OPEN ACCESS

Data-Driven Optimization for the Evolve-Filter-Relax Regularization of Convection-Dominated Flows

Anna Ivagnes[1] | Maria Strazzullo[2] | Michele Girfoglio[3] | Traian Iliescu[4] | Gianluigi Rozza[1]

[1]Mathematics Area, SISSA mathLab, Trieste, Italy | [2]Dipartimento di Scienze Matematiche "G. L. Lagrange" (DISMA), Politecnico di Torino, Torino, Italy | [3]Department of Engineering, University of Palermo, Palermo, Italy | [4]Department of Mathematics, Virginia Tech, Blacksburg, VA, USA

Correspondence: Maria Strazzullo (maria.strazzullo@polito.it)

Received: 25 March 2025 | **Revised:** 25 March 2025 | **Accepted:** 11 April 2025

Funding: This work was supported by Gruppo Nazionale per il Calcolo Scientifico (GNCS) (https://doi.org/10.13039/100017142); by H2020 Marie Skłodowska-Curie Actions under grant number 872442 (ARIA) (https://doi.org/10.13039/100010665); by the European Union grant 20227K44ME and FSE REACT-EU 2021; by iNEST-D.D. 1058 23/06/2022, ECS00000043; by ECCOMAS EYIC Grant; and by National Science Foundation grant DMS-2012253 (https://doi.org/10.13039/100000001).

Keywords: computational fluid dynamics | evolve-filter-relax | optimization | regularized models | turbulent flows

ABSTRACT

Numerical stabilization techniques are often employed in under-resolved simulations of convection-dominated flows to improve accuracy and mitigate spurious oscillations. Specifically, the evolve–filter–relax (EFR) algorithm is a framework that consists of evolving the solution, applying a filtering step to remove high-frequency noise, and relaxing through a convex combination of filtered and original solutions. The stability and accuracy of the EFR solution strongly depend on two parameters, the filter radius δ and the relaxation parameter χ. Standard choices for these parameters are usually fixed in time, and related to the full order model setting, that is, the grid size for δ and the time step for χ. The key novelties with respect to the standard EFR approach are: (i) time-dependent parameters $\delta(t)$ and $\chi(t)$, and (ii) data-driven adaptive optimization of the parameters in time, considering a fully-resolved simulation as reference. In particular, we propose three different classes of optimized-EFR (**Opt-EFR**) strategies, aiming to optimize one or both parameters. The new Opt-EFR strategies are tested in the under-resolved simulation of a turbulent flow past a cylinder at $Re = 1,000$. The Opt-EFR proved to be more accurate than standard approaches by up to 99%, while maintaining a similar computational time. In particular, the key new finding of our analysis is that such accuracy can be obtained only if the optimized objective function includes: (i) a *global* metric (as the kinetic energy), and (ii) *spatial gradients*' information.

1 | Introduction

It is well known that, in the convection-dominated (e.g., turbulent), under-resolved regime (i.e., when the number of degrees of freedom is not high enough to accurately represent the complex underlying dynamics), classical numerical discretizations of the Navier-Stokes equations (NSE) generally yield spurious numerical oscillations [1, 2]. To alleviate this inaccurate behavior, these discretizations are generally equipped with numerical stabilization, large eddy simulation [3, 4], or Reynolds-Averaged Navier–Stokes (RANS) models [1, 2, 5–9]. Regularized models (see [2] for a review) are a class of numerical stabilization strategies that leverage spatial filtering to increase the numerical stability of the underlying model and eliminate the spurious numerical oscillations. One of the most popular regularized models is the *evolve-filter-relax (EFR)* algorithm, which consists of the

---

INdAM research group GNCS member.

International Journal for Numerical Methods in Engineering, 2025; 126:e70042
https://doi.org/10.1002/nme.70042

following three steps: (i) *Evolve*: Use the NSE discretization to advance the velocity approximation at the current time step to an intermediate approximation at the next time step. (ii) *Filter*: Use a spatial filter to smooth the intermediate approximation in step (i). The goal of this step is to eliminate the spurious oscillations from the intermediate velocity approximation in step (i). (iii) *Relax*: Set the velocity approximation at the next time step to be a convex combination of the filtered velocity in step (ii) and the intermediate velocity approximation in step (i). The goal of this step is to prevent an overdiffusive velocity approximation, which could occur in step (ii) if, for example, the filter radius is too large. The EFR strategy has been successfully used in conjunction with classical numerical discretization techniques, ranging from finite element (FE), to spectral element, to finite volume methods, to reduced order modeling (see, e.g., [10–28]) in challenging applications, ranging from nuclear engineering design, to the quasi-geostrophic equations describing the large-scale ocean circulation, to cardiovascular modeling (see, e.g., [2, 17, 29–34]). The EFR's popularity is due to its accuracy and low computational cost. We emphasize, however, that probably the main reasons for the EFR's popularity in practical applications are its *ease of implementation*, *modularity*, and *minimally intrusive* character. Indeed, the EFR algorithm is embarrassingly easy to implement: Given access to the stiffness matrix of the underlying numerical discretization, the spatial filter (and, thus, the EFR algorithm) can be implemented in a matter of minutes. Furthermore, the EFR algorithm is modular and minimally intrusive: Given a *legacy* NSE code, one can very easily add a modular spatial filter subroutine and a modular relaxation step without having to modify the legacy code.

The EFR algorithm has been proven to be a popular and effective strategy in important applications. Its further development, however, currently faces a major roadblock: The EFR algorithm involves two parameters (i.e., the filter radius and the relaxation parameter) that need to be carefully calibrated in order to obtain accurate approximations. This parameter calibration is critical for the EFR's success: If the filter radius is too large, the EFR model is overdiffusive and the results are inaccurate (smeared) [35]. On the other hand, if the filter radius is too small, there is not enough dissipation, and the EFR model can actually blow up. Similarly, a relaxation parameter that is too small or too large can yield grossly inaccurate EFR results. Thus, finding the optimal filter radius and relaxation parameter is critical for the EFR's success. We note that the sensitivity of the EFR results to the model parameters has been well documented (see, e.g., the numerical investigations in References [17, 25, 27, 35, 36]). We also note that the importance of the EFR's parameter calibration is not a surprise since parameter sensitivity is a central issue for most numerical stabilizations [1, 2, 37, 38].

Recognizing the importance of finding the optimal filter radius and relaxation parameter in the EFR model, several approaches have been proposed early on. For example, in References [17, 25], the EFR filter radius was chosen to be the mesh size of the underlying FE discretization. Another choice was proposed in Reference [36], where the EFR filter radius was chosen to be the Kolmogorov lengthscale. Similarly, several choices for the EFR relaxation parameter have been proposed. For example, choosing the relaxation parameter proportional to the time step is a popular choice in the EFR literature [25, 36]. We note, however, that heuristic arguments have been used to motivate higher values for the relaxation parameter [24, 25, 36].

We emphasize that the vast majority of the current choices for the EFR filter radius and relaxation parameter yield *constant* values in both space and time for these two parameters (an important exception is the nonlinear filtering proposed in Reference [17].) Using constant values for the parameters imposes significant limitations on the EFR's performance. For example, if the flow transitions from a laminar regime to a turbulent regime, the EFR's filter radius and/or time relaxation parameter should increase to adapt to this transition. We emphasize that this is simply not possible with the current parameter choices.

The key novelties of this contribution are:

i. the introduction of time-dependent EFR parameters;

ii. computing the optimal parameters at different time instances with an adaptive *data-driven optimization* strategy, leveraging the available reference data;

iii. the inclusion of *global* quantities into the objective function, such as the kinetic energy; and

iv. the inclusion of *spatial gradients* into the objective function.

The proposed strategies are tested in the marginally-resolved simulation of a flow past a cylinder at a Reynolds number $Re = 1,000$.

We emphasize that our new strategy is fundamentally different from the current approaches:

- This is the first time that data-driven modeling is leveraged to determine the EFR parameters.
- Furthermore, the new data-driven parameters vary in time, which is in stark contrast to the vast majority of current methods used to determine the EFR parameters.

The rest of the paper is organized as follows: In Section 2, we present the EFR strategy and discuss the critical role played by the EFR parameters, the filter radius, and the relaxation parameter. In Section 3, we introduce the **Opt-EFR** approaches, novel data-driven optimization algorithms for the EFR strategy. Specifically, we introduce three Opt-EFR strategies:

(i) In the χ-Opt-EFR approach, we fix the filter radius and optimize the relaxation parameter. (ii) In the δ-Opt-EFR approach, we fix the relaxation parameter and optimize the filter radius. (iii) Finally, in the $\delta\chi$-Opt-EFR approach, we optimize both the relaxation parameter and the filter radius.

In Section 4, we perform a numerical investigation of the three new data-driven EFR strategies in the numerical simulation of incompressible flow past a circular cylinder at Reynolds number $Re = 1,000$ on a relatively coarse mesh. For comparison purposes, we also consider the standard EFR approach. As a benchmark for our numerical investigation, we use a DNS on a fine mesh. Finally, in Section 5 we draw conclusions and outline future research directions.

2 | The EFR Algorithm

This Section is dedicated to a review of the state of the art related to the methodology we investigate. In particular, we briefly recall the discretized Navier–Stokes formulation and the Evolve-Filter-Relax (EFR) approach in Section 2.1. For more details, we refer the reader to [18, 25, 36, 39]. Moreover, in Section 2.2, we highlight the bottleneck of the EFR formulation, and the motivations to enhance it by optimization.

2.1 | The Navier–Stokes Equations and the EFR Algorithm

As a mathematical model, we consider the **incompressible Navier–Stokes equations (NSE)**. Given a fixed domain $\Omega \subset \mathbb{R}^D$, with $D = 2, 3$, we consider the motion of an incompressible fluid having velocity $\boldsymbol{u} \doteq \boldsymbol{u}(\boldsymbol{x}, t) \in \mathbb{U}$ and pressure $p \doteq p(\boldsymbol{x}, t) \in \mathbb{Q}$ whose dynamics is represented by the incompressible NSE:

$$\begin{cases} \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla p = 0 & \text{in } \Omega \times (t_0, T), \\ \nabla \cdot \boldsymbol{u} = 0 & \text{in } \Omega \times (t_0, T), \\ \boldsymbol{u} = \boldsymbol{u}_D & \text{on } \partial\Omega_D \times (t_0, T), \\ -p\ \boldsymbol{n} + \nu \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} = 0 & \text{on } \partial\Omega_N \times (t_0, T) \end{cases} \tag{1}$$

endowed with the initial condition $\boldsymbol{u} = \boldsymbol{u}_0$ in $\Omega \times \{t_0\}$, where $\partial\Omega_D \cup \partial\Omega_N = \partial\Omega$, $\partial\Omega_D \cap \partial\Omega_N = \emptyset$, ν is the kinematic viscosity, and $\mathbb{U}$ and $\mathbb{Q}$ are suitable Hilbert function spaces. The functions $\boldsymbol{u}_D$ and $\boldsymbol{u}_0$ are given. The flow regime is defined by the adimensional **Reynolds number**

$$Re \doteq \frac{U\,L}{\nu} \tag{2}$$

where U and L represent the characteristic velocity and length scales of the system, respectively. When the Reynolds number is large, the inertial forces dominate the viscous forces: this setting is generally referred to as the convection-dominated regime.

In the numerical tests, we employ the **Finite Element Method (FEM)** and a backward differentiation formula of order 1 (BDF1) for space and time discretization, respectively. In under-resolved or marginally-resolved numerical simulations, namely when employing a coarse grid to run the FEM simulations, standard spatial discretization techniques yield spurious numerical oscillation in convection-dominated regimes.

To alleviate this behavior, we equip the simulation with the **Evolve-Filter-Relax (EFR)** algorithm. To introduce the notation, we consider a time step Δt. Let $t^n = t_0 + n\Delta t$ for $n = 0, \dots, N_T$, and $T = t_0 + N_T \Delta t$. We denote the semi-discrete FEM velocity and pressure at time t^n with $\boldsymbol{u}^n \in \mathbb{U}^{N_h^{\boldsymbol{u}}}$ and $p^n \in \mathbb{Q}^{N_h^p}$, respectively, where $N_h^{\boldsymbol{u}}$ and N_h^p are their degrees of freedom, respectively. The EFR algorithm at time t^{n+1} reads:

$$\textbf{(I)}\quad \text{Evolve}: \begin{cases} \frac{\boldsymbol{w}^{n+1} - \boldsymbol{u}^n}{\Delta t} + (\boldsymbol{w}^{n+1} \cdot \nabla)\boldsymbol{w}^{n+1} + \\ -\nu \Delta \boldsymbol{w}^{n+1} + \nabla p^{n+1} = 0 & \text{in } \Omega \times \{t_{n+1}\}, \\ \nabla \cdot \boldsymbol{w}^{n+1} = 0 & \text{in } \Omega \times \{t_{n+1}\}, \\ \boldsymbol{w}^{n+1} = \boldsymbol{u}_D^{n+1} & \text{on } \partial\Omega_D \times \{t_{n+1}\}, \\ -p^{n+1} \cdot \boldsymbol{n} + \frac{\partial \boldsymbol{w}^{n+1}}{\partial \boldsymbol{n}} = 0 & \text{on } \partial\Omega_N \times \{t_{n+1}\}. \end{cases}$$

$$\textbf{(II)}\quad \text{Filter}: \begin{cases} -2\delta^2\ \Delta \overline{\boldsymbol{w}}^{n+1} + \overline{\boldsymbol{w}}^{n+1} = \boldsymbol{w}^{n+1} & \text{in } \Omega \times \{t_{n+1}\}, \\ \overline{\boldsymbol{w}}^{n+1} = \boldsymbol{u}_D^{n+1} & \text{on } \partial\Omega_D \times \{t_{n+1}\}, \\ \frac{\partial \overline{\boldsymbol{w}}^{n+1}}{\partial \boldsymbol{n}} = 0 & \text{on } \partial\Omega_N \times \{t_{n+1}\}. \end{cases}$$

$$\textbf{(III)}\quad \text{Relax}:\ \boldsymbol{u}^{n+1} = (1 - \chi)\ \boldsymbol{w}^{n+1} + \chi\ \overline{\boldsymbol{w}}^{n+1}$$

for $\chi \in [0, 1]$ relaxation parameter. We denote by $\boldsymbol{w}$ and $\overline{\boldsymbol{w}}$ the evolved and the filtered velocity, respectively. Solving the first step (I) with $\boldsymbol{w}^{n+1} = \boldsymbol{u}^{n+1}$ equals solving the standard NSE. In step (II), we employ a *differential filter* (DF) filter radius δ (i.e., the spatial lengthscale over the spatial filter takes action). The success of DFs is due to their mathematical robustness [1]. Indeed, the DF leverages an elliptic operator and eliminates the small scales (i.e., high frequencies) from the input velocity, smoothing it and acting as a spatial filter. Step (III) is a relaxation step: A convex combination of the evolved and filtered velocity is performed at a new time step. In particular, the extreme cases are:

i. $\chi = 1$: We only retain the filtered solution $\boldsymbol{u}^{n+1} = \overline{\boldsymbol{w}}^{n+1}$;

ii. $\chi = 0$: The final solution is exactly the solution of the incompressible NSE (I), that is, no filtering contribution is included and $\boldsymbol{u}^{n+1} = \boldsymbol{w}^{n+1}$.

In other words, the relaxation parameter χ is used to regulate the amount of the filter contribution [12, 14, 20], to increase the accuracy of possible over-diffusive results of Step (II), as shown for example in Reference [36]. We remark that we call *noEFR simulation* the numerical solution of the discretized Navier–Stokes equations with no regularization applied, following the notation of [25].

2.2 | On the Role of the EFR Parameters

A natural question arises in this setting: *What are the optimal choices for* δ and χ? In the literature, some scale arguments have been proposed (we refer the reader to [25, 36] and the references therein for more details on the topic):

- Common choices for the filter radius δ are the minimum mesh-scale diameter h_{min} or the Kolmogorov length scale $\eta = L \cdot Re^{\frac{3}{4}}$. Another choice is the energy-based lengthscale [40].
- For the relaxation, the scaling $\chi \sim \Delta t$ is commonly used [12]. However, its choice highly depends on the specific test case taken into account, and higher values have been employed following heuristic arguments in References [17, 25, 36].

While the choice of the filter radius is related to physical quantities, the choice of the relaxation parameter is based on experience, and it might completely change with respect to the test case considered. In this context, many adjustments and improvements can be made. For example, time-dependent flow features can be better captured with a time-dependent parameter $\chi(t)$, or the differential filter may have local action [17]. Clearly, *The main bottleneck of the EFR procedure relies upon tuning the filter radius and the relaxation parameters.*

Indeed, in the standard EFR algorithm, the parameters' values need to be carefully tuned in a pre-processing stage to improve the accuracy of the solution:

i. If the values of δ and/or χ are **too large**, it may happen that the solution is **over-diffusive** and it does not manifest any turbulent behavior.

ii. If the values of δ and/or χ are **too small**, the filter action may not be adequate, leading to **spurious oscillations**, as happens in the noEFR scenario.

Moreover, as mentioned above, the flow dynamics may not be properly captured considering parameters constant in time, especially when dealing with turbulent flows or with test cases including a laminar-to-turbulent transition. For instance, we may want larger filter parameters (hence, higher δ and/or χ) in the laminar-to-turbulent transition, as well as in fully turbulent regimes.

For this purpose, having time-dependent parameters would help in better capturing the flow dynamics, avoiding, for instance, overdiffusion phenomena. In this manuscript, **for the first time, we introduce a data-driven strategy** to compute the time-dependent parameters $\delta(t)$ and $\chi(t)$, which allow us to better capture the flow dynamics in unsteady turbulent regimes. In this framework, we aim at investigating different **data-driven optimization algorithms**, with the ultimate goal of developing *an adaptive strategy* to optimize $\delta(t)$ and $\chi(t)$.

3 | The Data-Driven Optimized EFR Algorithms

In this section, we aim at obtaining *optimal values* for the EFR parameters, enhancing standard approaches by adaptive optimization algorithms. In this way, we increase the accuracy of the reg-simulation with respect to a Direct Numerical Simulation (DNS) simulation taken as a reference.

Our investigation goes beyond constant parametric values and focuses on three specific Opt-EFR strategies. Specifically, calling $\boldsymbol{\mu}^n = \boldsymbol{\mu}(t^n)$ the parameter(s) optimized at each time step t^n, the three approaches are:

1. χ-**Opt-EFR**, aimed to find the optimal $\chi(t)$ for fixed $\delta = \eta$, hence $\boldsymbol{\mu}^n = \chi^n$;
2. δ-**Opt-EF**, aimed to find the optimal $\delta(t)$, with no relaxation, namely $\chi = 1$, hence $\boldsymbol{\mu}^n = \delta^n$;
3. $\delta\chi$-**Opt-EFR**, aimed to find the optimal $\delta(t)$ and $\chi(t)$, hence $\boldsymbol{\mu}^n = (\delta^n, \chi^n)$.

The presented optimization procedures are based on a data-driven algorithm trained taking as reference some pre-computed data. In our specific setting, the training data are represented by the reference velocity and pressure fields of a simulation in a resolved regime.

We refer to those fields as $\boldsymbol{u}_{ref} \in \mathbb{U}^{M_h^{\boldsymbol{u}}}$ and $p_{ref} \in \mathbb{Q}^{M_h^p}$, where $M_h^{\boldsymbol{u}} \gg N_h^{\boldsymbol{u}}$ and $M_h^p \gg N_h^p$. Such data are provided by a simulation over a refined mesh yielding $M_h^{\boldsymbol{u}} + M_h^p$ degrees of freedom (dofs). The regularization, instead, is applied to a coarser mesh with respect to the DNS one, with $N_h^{\boldsymbol{u}} + N_h^p$ dofs. Moreover, it is important to remark on the **generality** and the **modularity** of the proposed algorithms, which can be applied to any experimental setting. The motivations behind the choices of the test cases of this contribution are postponed to the numerical sections.

A schematic comparison between the standard EFR and the Opt-EFR methods is displayed in Figure 1. The standard EFR is characterized by a standard time-stepping procedure, as described in Section 2.1 and shown in Figure 1a. On the other hand, the Opt-EFR algorithms perform an additional optimization stage to find the optimal $\delta(t)$ and/or $\chi(t)$, as highlighted in Figure 1b.

3.1 | The Objective Functions

In the optimization algorithms, we will consider either *local* or *global* contributions. Specifically, the *local* approach refers to a pointwise evaluation and minimization of the discrepancy between the reference and the predicted quantity. On the other hand, the *global* approach consists of the evaluation and minimization of the discrepancy of the global energy.

We refer to *local* quantities if we minimize the mean squared error (MSE) between the DNS reference solution (projected on the coarse mesh) and the updated EFR solution.

The local objective function can then be written as follows:

$$\mathcal{L}_{\text{local}}(\boldsymbol{\mu}^n) = w_{\boldsymbol{u}}\ \mathcal{L}^{\boldsymbol{u}}_{\text{local}}(\boldsymbol{\mu}^n) + w_{\nabla \boldsymbol{u}}\ \mathcal{L}^{\nabla \boldsymbol{u}}_{\text{local}}(\boldsymbol{\mu}^n) \tag{3}$$

where the individual contributions are:

$$\begin{aligned} \mathcal{L}^{\boldsymbol{u}}_{\text{local}}(\boldsymbol{\mu}^n) &= \text{MSE}\Big(\boldsymbol{u}^n(\boldsymbol{\mu}^n), \boldsymbol{u}^n_{ref}\Big) \\ \mathcal{L}^{\nabla \boldsymbol{u}}_{\text{local}}(\boldsymbol{\mu}^n) &= \text{MSE}\Big(\nabla \boldsymbol{u}^n(\boldsymbol{\mu}^n), \nabla \boldsymbol{u}^n_{ref}\Big) \end{aligned} \tag{4}$$

with $w_{\boldsymbol{u}}$ and $w_{\nabla \boldsymbol{u}}$ scalar quantities. As pointed out in the results in Section 4.1, the local contributions may not be appropriate metrics to optimize in this test case. In particular, the Opt-EFR results and the DNS may differ a lot in terms of qualitative and quantitative behaviors, even if the local MSE is relatively small. For this reason, we investigate the following global objective function:

$$\mathcal{L}_{\text{global}}(\boldsymbol{\mu}^n) = w_{\boldsymbol{u}}\mathcal{L}^{\boldsymbol{u}}_{\text{global}}(\boldsymbol{\mu}^n) + w_{\nabla \boldsymbol{u}}\mathcal{L}^{\nabla \boldsymbol{u}}_{\text{global}}(\boldsymbol{\mu}^n) + w_p\mathcal{L}^{p}_{\text{global}}(\boldsymbol{\mu}^n) \tag{5}$$

where the global contributions are:

$$\begin{aligned} \mathcal{L}^{\boldsymbol{u}}_{\text{global}}(\boldsymbol{\mu}^n) &= \left| \frac{\|\boldsymbol{u}^n(\boldsymbol{\mu}^n)\|^2_{L^2(\Omega)} - \|\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}}{\|\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}} \right| \\ \mathcal{L}^{\nabla \boldsymbol{u}}_{\text{global}}(\boldsymbol{\mu}^n) &= \left| \frac{\|\nabla \boldsymbol{u}^n(\boldsymbol{\mu}^n)\|^2_{L^2(\Omega)} - \|\nabla \boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}}{\|\nabla \boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}} \right| \\ \mathcal{L}^{p}_{\text{global}}(\boldsymbol{\mu}^n) &= \left| \frac{\|p^{n+1}(\boldsymbol{\mu}^n)\|^2_{L^2(\Omega)} - \|p^{n+1}_{ref}\|^2_{L^2(\Omega)}}{\|p^{n+1}_{ref}\|^2_{L^2(\Omega)}} \right| \end{aligned} \tag{6}$$

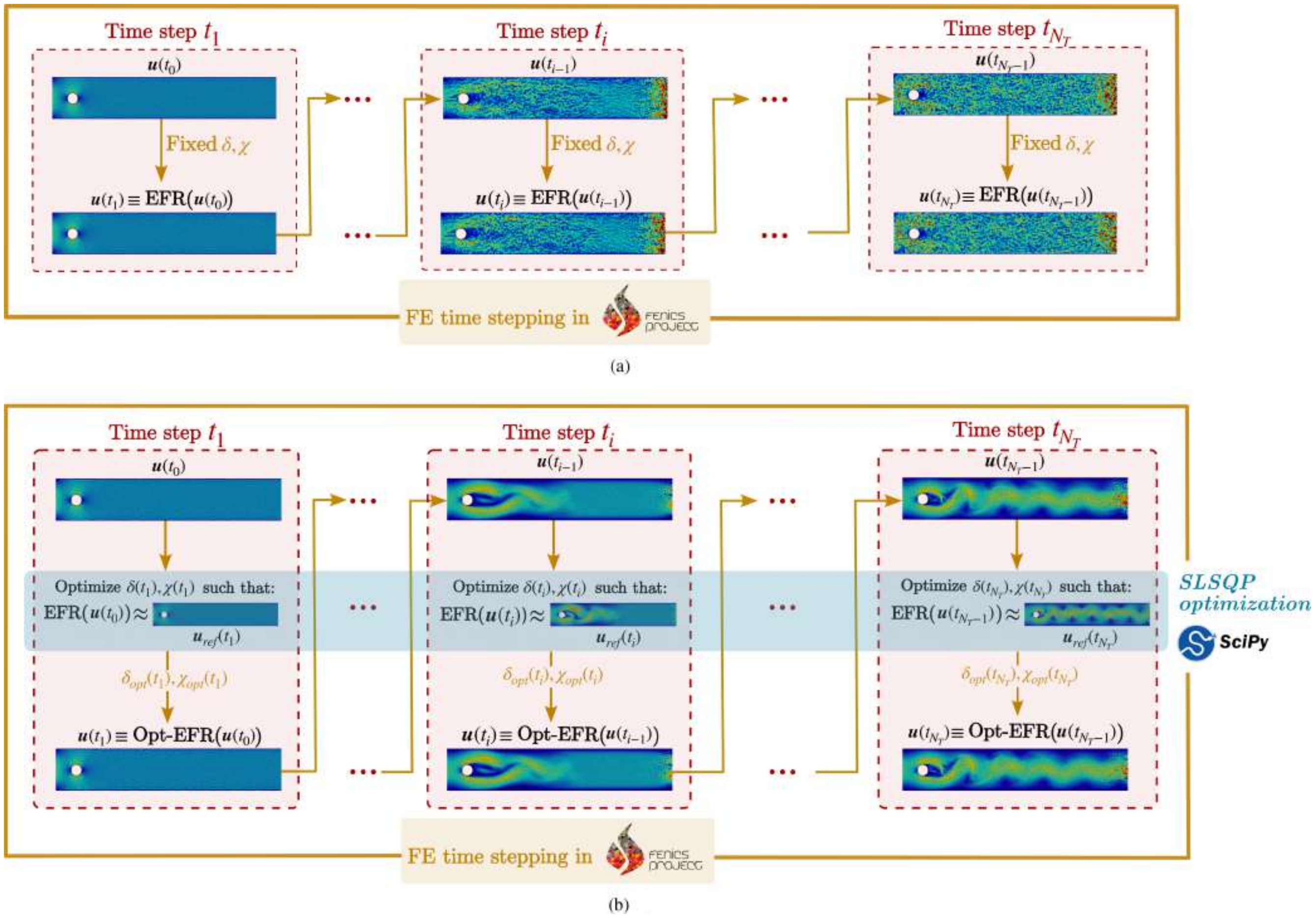

Time step t_1
$\boldsymbol{u}(t_0)$
Fixed δ, χ
$\boldsymbol{u}(t_1) \equiv \mathrm{EFR}(\boldsymbol{u}(t_0))$
Time step t_i
$\boldsymbol{u}(t_{i-1})$
Fixed δ, χ
$\boldsymbol{u}(t_i) \equiv \mathrm{EFR}(\boldsymbol{u}(t_{i-1}))$
Time step t_{N_T}
$\boldsymbol{u}(t_{N_T-1})$
Fixed δ, χ
$\boldsymbol{u}(t_{N_T}) \equiv \mathrm{EFR}(\boldsymbol{u}(t_{N_T-1}))$
FE time stepping in
FEniCS PROJECT
(a)
Time step t_1
$\boldsymbol{u}(t_0)$
Optimize $\delta(t_1), \chi(t_1)$ such that:
$\mathrm{EFR}(\boldsymbol{u}(t_0)) \approx$
$\boldsymbol{u}_{ref}(t_1)$
$\delta_{opt}(t_1), \chi_{opt}(t_1)$
$\boldsymbol{u}(t_1) \equiv \text{Opt-EFR}(\boldsymbol{u}(t_0))$
Time step t_i
$\boldsymbol{u}(t_{i-1})$
Optimize $\delta(t_i), \chi(t_i)$ such that:
$\mathrm{EFR}(\boldsymbol{u}(t_i)) \approx$
$\boldsymbol{u}_{ref}(t_i)$
$\delta_{opt}(t_i), \chi_{opt}(t_i)$
$\boldsymbol{u}(t_i) \equiv \text{Opt-EFR}(\boldsymbol{u}(t_{i-1}))$
Time step t_{N_T}
$\boldsymbol{u}(t_{N_T-1})$
Optimize $\delta(t_{N_T}), \chi(t_{N_T})$ such that:
$\mathrm{EFR}(\boldsymbol{u}(t_{N_T-1})) \approx$
$\boldsymbol{u}_{ref}(t_{N_T})$
$\delta_{opt}(t_{N_T}), \chi_{opt}(t_{N_T})$
$\boldsymbol{u}(t_{N_T}) \equiv \text{Opt-EFR}(\boldsymbol{u}(t_{N_T-1}))$
SLSQP optimization
SciPy
FE time stepping in
FEniCS PROJECT
(b)

FIGURE 1 | Flowchart of the Opt-EFR and standard EFR approaches. (a) Standard EFR, (b) Opt-EFR.

TABLE 1 | Acronyms and main features of the Opt-EFR algorithms. Gray cells indicate that the algorithm does not present that feature.

	Objective function type		**Contributions in objective function**		
Acronym	**local**	**global**	**$\boldsymbol{u}$ discrepancy**	**$\nabla\boldsymbol{u}$ discrepancy**	**p discrepancy**
Opt-EFR$^{\boldsymbol{u}}_{\text{local}}$	✓	▨	✓	▨	▨
Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{local}}$	✓	▨	✓	✓	▨
Opt-EFR$^{\boldsymbol{u}}_{\text{global}}$	▨	✓	✓	▨	▨
Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$	▨	✓	✓	✓	▨
Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$	▨	✓	✓	✓	✓

for w_p scalar quantity. In particular, we set $w_{\boldsymbol{u}} \neq 0$ and $w_{\nabla\boldsymbol{u}} = w_p = 0$ in algorithms of type $(\cdot)^{\boldsymbol{u}}_{\text{global}}$, $w_{\boldsymbol{u}} \neq 0$, $w_{\nabla\boldsymbol{u}} \neq 0$, and $w_p = 0$ in $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ algorithms, while all the contributions are activated in $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$.

An overview of the methods investigated in this work, depending on the different loss functionals considered, is provided in Table 1.

It is worth highlighting that the expression of $\mathcal{L}^{p}_{\text{global}}$ in Equation (6) penalizes the pressure obtained as the solution of the evolve step at the time instance t^{n+1}. Therefore, the optimization algorithms penalizing the pressure contribution—namely, $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$—are computationally more expensive than the others, since a further evolve step must be performed for each evaluation.

Indeed, in Steps (II) and (III) at time t^n apply only to the velocity field. Thus, the pressure contribution in Equation (6) needs to be computed considering the pressure at time t^{n+1}, which is the first pressure field affected by the optimization of $\boldsymbol{\mu}$ at step t^n. We investigate the effect of the pressure contribution only in the *global* approach because the numerical results for the *local* method are not accurate (as shown in Section 4).

For additional details on the three classes χ-Opt-EFR, δ-Opt-EF, and $\delta\chi$-Opt-EFR, we refer the reader to the supplementary Section A.

Remark 1. It is worth highlighting that the scalar weights $w_{\boldsymbol{u}}$, $w_{\nabla \boldsymbol{u}}$, and w_p are taken exactly equal to 1 when they are different from 0, namely when the corresponding terms are activated in the objective functions (3) and (5). Other choices can be made, but finding the optimal choices for the loss weights was beyond the goal of the manuscript.

3.2 | The Optimization Process

The algorithms in Table 1 perform at each time instance Step (I), and look for an optimal time-dependent $\delta(t)$ and/or $\chi(t)$ to be employed in Step (II) and/or (III).

The process is summarized in Algorithm 1, where $y|\mathbb{Y}$ denotes the projection of the field y onto the space $\mathbb{Y}$.

Algorithm 1 presents the general pseudocode that can be applied for all the algorithms presented in Table 1, by replacing the objective function $\mathcal{L}(\chi^{n+1})$ with the specific functional of the proposed algorithm.

Moreover, we employ a *sequential least squares programming* (SLSQP) algorithm [41, 42], which is an iterative method for constrained nonlinear optimization.

The only considered constraint is $\boldsymbol{\mu}^{n+1} \in \mathrm{A}$, where A is the admissibility range of the parameter(s).

In particular, we consider:

1. $\chi^{n+1} \in [0, 1]$ in χ-Opt-EFR algorithms, namely $\mathrm{A} = [0, 1]$;
2. $\delta^{n+1} \in [0.019\eta, 1.9\eta]$ in δ-Opt-EF algorithms, namely $\mathrm{A} = [0.019\eta, 1.9\eta]$; and
3. $\delta^{n+1} \in [0.019\eta, 1.9\eta]$ and $\chi^{n+1} \in [0, 1]$ in $\delta\chi$-Opt-EFR algorithms, namely $\mathrm{A} = [0.019\eta, 1.9\eta] \times [0, 1]$;

By adding the constraint $\delta^{n+1} \in [0.019\eta, 1.9\eta]$, the optimization problem seeks the optimal parameter within the range spanning an order of magnitude lower to one higher than the Kolmogorov length scale. This window has been established after a preliminary sensitivity analysis on the parameter δ, as it provides more physical results. The unconstrained optimization may indeed converge to values that are too large and far from the standard literature choices.

The underlined row (row 10 in the pseudocode 1) is considered only in the Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ algorithms, since it performs an additional evolve step to evaluate the pressure at the following time instance p^{n+2}. As an additional consideration, we consider as the initial value of the parameter(s) the common choice in the literature [25], namely $\delta^0 = \eta$ for the filter radius, and $\chi_0 = K\Delta t$, with $K = 5$, for the relaxation parameter.

Remark 2. It is worth specifying that the parameters are not optimized every time step. Instead, the time window between two consecutive optimizations is $\Delta t_{opt} > \Delta t$. Specifically, $\Delta t_{opt} = k\Delta t$, with $k \in \{10, 20, \dots, 90, 100\}$.

Every Δt_{opt}, the algorithm takes into account the updated solution of the system and therefore needs to be trained on the fly. The parameter is fixed to its latest optimal value in the time window between two subsequent optimization steps. In the numerical results, we will also evaluate how the optimization step influences the accuracy and efficiency of the solution.

3.2.1 | Software details

The EFR method is implemented using the finite-element Python-based open-source library FEniCS [43]. We perform the

ALGORITHM 1 | Pseudocode for algorithms Opt-EFR optimizing both parameters. Row 9 (underlined) is considered only for algorithms of type Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$. The optimization part is highlighted in gray.

```
1:  $\boldsymbol{u}^n_{ref}$, $n = 0, \dots, N_T - 1$; $N^u_h$; $\boldsymbol{u}^0$, $\Delta t_{opt}$, $\boldsymbol{\mu}^0$, A range of parameter(s).   ▷ Inputs needed
2:  for $n \in [0, \dots, N_T - 1]$ do
3:      (I) $\rightarrow \boldsymbol{w}^{n+1}$                                   ▷ Perform Evolve step
4:      $\boldsymbol{u}^{n+1}_{ref} \leftarrow \boldsymbol{u}^{n+1}_{ref} | \mathbb{U}^{N^u_h}$      ▷ Project the reference field into the coarse mesh
5:      $\boldsymbol{\mu}^{n+1} \leftarrow \boldsymbol{\mu}^n$                              ▷ Initialize parameter(s)
6:      if $\mathrm{mod}(n, \Delta t_{opt}) = 0$ then                ▷ Perform optimization every $\Delta t_{opt}$
7:          $iter = 0$
8:          repeat
9:              (II) + (III) $\rightarrow \boldsymbol{u}^{n+1}$                      ▷ Filter and Relax step
10:             $\underline{\text{(I)} \rightarrow p^{n+2}}$                        ▷ $\underline{\text{(Second) Evolve step}}$
11:             Optimize $\mathcal{L}(\boldsymbol{\mu}^{n+1})$, subject to $\boldsymbol{\mu}^{n+1} \in \mathrm{A}$
12:             $iter \leftarrow iter + 1$
13:             Output current solution: $\boldsymbol{\mu}^{n+1}$
14:         until Convergence or maximum iterations reached
15:     end if
16:     Solve (II) + (III) with the optimized $\boldsymbol{\mu}^{n+1} \rightarrow$ Update $\boldsymbol{u}^{n+1}$   ▷ Perform Filter and Relax
17: end for
```

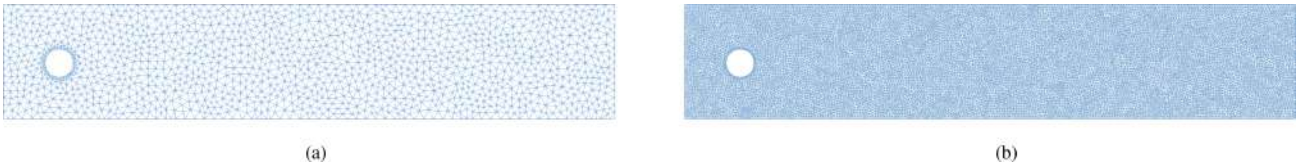


FIGURE 2 | Meshes taken into account to run the simulations. (a) Coarse mesh, (b) Fine mesh.

optimization algorithms using the open-source Python-based library Scipy [44].

4 | Numerical Results

This section shows the numerical results obtained by applying the methods proposed in Sections 2 and 3. We consider the test case of the incompressible flow past a circular cylinder at Reynolds number $Re = 1,000$, constant in time.

We consider as reference in the optimization algorithms a DNS on a fine mesh, and our goal is to minimize the discrepancy between the optimized-EFR fields and the reference DNS fields.

Remark 3. In this project, we call DNS the reference simulation considered to train the proposed data-driven algorithm. However, our reference cannot be considered a fully-resolved simulation in its classical sense. Indeed, in Figure 4a,b show that the velocity fields still exhibit spurious numerical oscillations.

The two meshes taken into account are displayed in Figure 2.

The coarse mesh is characterized by $N_h^{\boldsymbol{u}} = 12430$ dofs for the velocity field and $N_h^p = 1623$ for the pressure. On the other hand, the fine mesh is characterized by $M_h^{\boldsymbol{u}} = 100676$ and $M_h^p = 12760$. We perform the simulations in the time interval $[0, 4]$, and the adaptive optimization algorithms described in Section 3 are tested in the reconstructive regime.

The results obtained by the EFR simulations are compared with the DNS results using the following error metrics:

- the L^2 relative errors

$$E_{L^2}^{u}(t_n) = \frac{\|\boldsymbol{u}^n - \boldsymbol{u}_{ref}^n\|_{L^2(\Omega)}}{\|\boldsymbol{u}_{ref}^n\|_{L^2(\Omega)}}, \quad E_{L^2}^{p}(t_n) = \frac{\|p^n - p_{ref}^n\|_{L^2(\Omega)}}{\|p_{ref}^n\|_{L^2(\Omega)}} \tag{7}$$

- the velocity relative error in the H^1 norm:

$$E_{H^1}^{u}(t_n) = \frac{\|\boldsymbol{u}^n - \boldsymbol{u}_{ref}^n\|_{H^1(\Omega)}}{\|\boldsymbol{u}_{ref}^n\|_{H^1(\Omega)}} \tag{8}$$

The domain is $\Omega \doteq \{(x, y) \in [0, 2.2] \times [0, 0.41]$ such that $(x - 0.2)^2 + (y - 0.2)^2 \geq 0.05^2\}$, and it is represented in Figure 3. We consider no-slip boundary conditions on $\partial\Omega_{wall} = \partial\Omega_B \cup \partial\Omega_T \cup \partial\Omega_C$ (blue solid line in Figure 3), and an inlet velocity $\mathbf{u}_{in}$ on $\partial\Omega_{in}$ (gold dashed line in Figure 3), given by:

$$\boldsymbol{u}_{in} \doteq \left(\frac{6}{0.41^2} y(0.41 - y), 0\right) \tag{9}$$

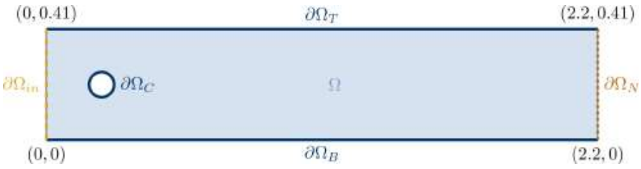


FIGURE 3 | The domain considered in the test cases.

On the outlet of the domain $\partial\Omega_N$ (brown dotted line in Figure 3), we employ standard free-flow conditions. The initial condition is $\boldsymbol{u}_0 = (0, 0)$. The viscosity is set to $\nu = 1 \times 10^{-4}$ and, consequently, the Reynolds number is $Re = 1,000$.

For the test case described above, not applying regularization yields an inaccurate field representation. We recall that, for the sake of the presentation, when no regularization is applied, we say that noEFR is performed, following the acronyms introduced in Reference [25].

Some preliminary results motivating our analysis are available in Figure 4, which yields the following conclusions:

- The noEFR simulations present spurious oscillations, as can be seen in plots (C) and (D). This motivates the use of the EFR algorithm, see, also [24, 35].
- The challenges related to this specific test case are clear even from the DNS simulations that, despite the use of a finer mesh, still present some numerical oscillations near the outlet domain, see plots (A) and (B).
- Moreover, the EFR method with standard choices $\delta = \eta = 5.62 \times 10^{-4}$ and $\chi = 5\Delta t$ yields poor results, as well. Indeed:
  - The standard EF approach is overdiffusive and does not capture the vortex shedding in the region near the cylinder, as can be seen from plots (G) and (H).
  - The standard EFR method yields spurious oscillations, meaning that the relaxation parameter χ is too small and should be carefully tuned, as shown in plots (E) and (F).

The above statements motivate the following analysis of parameter optimization, whose main goal is to improve the results of the noEFR and standard EFR approaches with adaptive optimization. In particular:

- Section 4.1 shows the results for χ-Opt-EFR;
- Section 4.2 shows the results for δ-Opt-EF;
- Section 4.3 shows the results for $\delta\chi$-Opt-EFR.

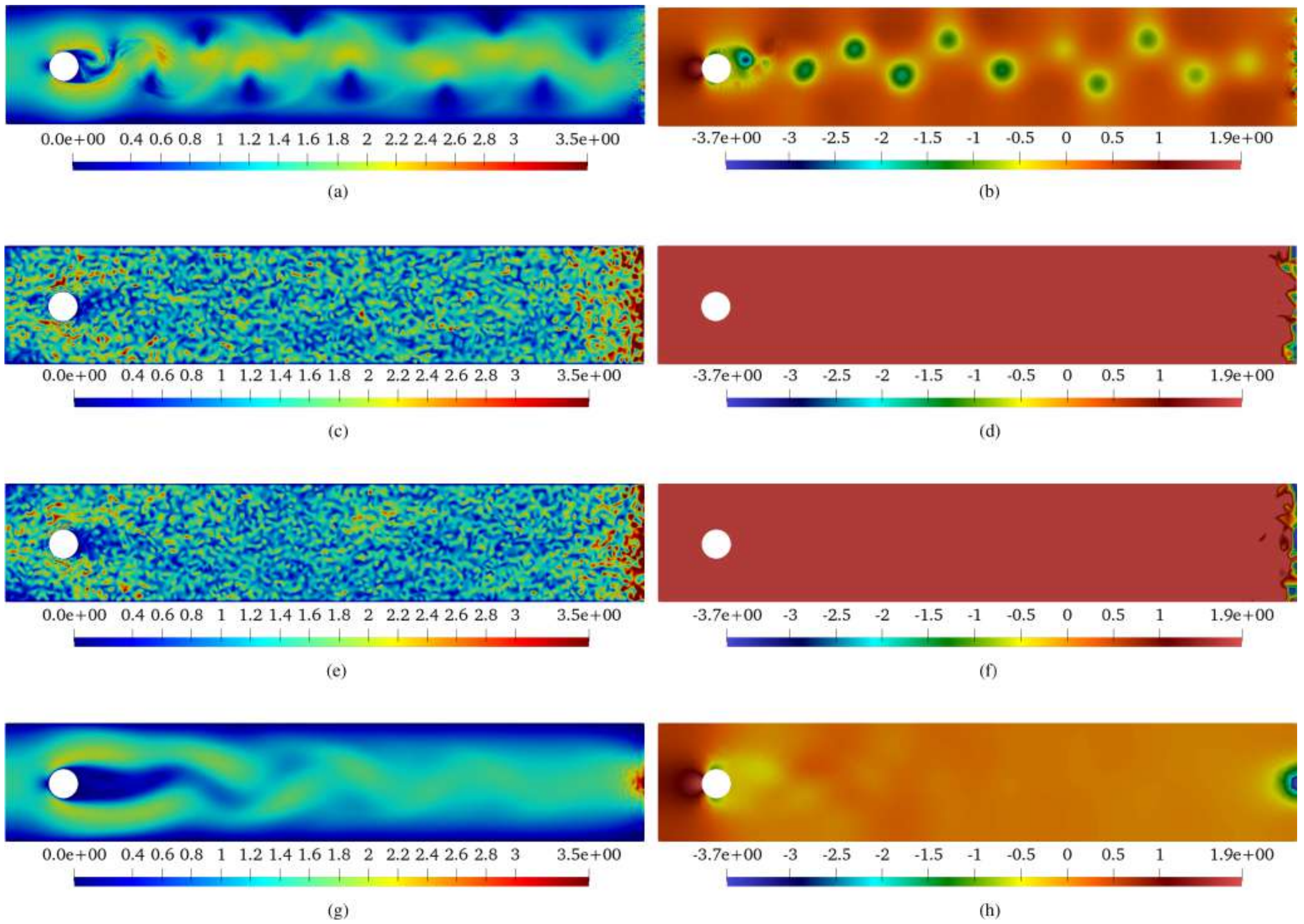


FIGURE 4 | From top to bottom: Velocity (left plots) and pressure (right plots) fields at final $t = 4$ for DNS on the fine mesh (the reference solution), and noEFR method on the coarse mesh (i.e., the solution to improve with regularization). The results of standard EFR and EF approaches on the coarse mesh are also displayed. (a) DNS on fine mesh-velocity, (b) DNS on fine mesh-pressure, (c) noEFR-velocity, (d) noEFR-pressure, (e) Standard EFR ($\delta = \eta$, $\chi = 5\Delta t$)-velocity, (f) Standard EFR ($\delta = \eta$, $\chi = 5\Delta t$)-pressure, (g) Standard EF ($\delta = \eta$)-velocity, and (h) Standard EF ($\delta = \eta$)-pressure.

4.1 | Numerical Results for χ-Opt-EFR

First, in Figure 5a, we include the $\chi(t)$ trends obtained with the different local or global objective functions taken into account.

When minimizing the **local** MSE discrepancy, the optimal $\chi(t)$ is close to 1 for each optimization time step. This means that the algorithm activates for almost 100% of the filtered velocity, and hence we expect an over-diffusive behavior, as happens in the standard EF approach.

On the other hand, when we only consider the **global** velocity norm discrepancy, the simulation blows up at $t = 1.6$, as can be seen in Figure B1, and in Figures 6 and 7d.

The simulation is more stable when the velocity gradients and the pressure contributions are also included in the objective function (approaches χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$). However, the last two plots on the right in Figure 5a show an oscillating and chaotic behavior of $\chi(t)$. To better understand the distribution of the parameters' values, we present in Figure 5b the relative number of optimal χ^n values attained by the Opt-EFR algorithms, with $n = 1, \ldots, N_T$. In particular, for a given value, n_{points} is the number of times χ^n attains the given value, while n_{total} is the total number of samples, namely 1,000. Figure 5 shows us that the number of $\chi(t)$ elements close to 1 is around 25% in both algorithms χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$. This means the optimization algorithm converges statistically more often to values close to 1, deactivating the relaxation step.

A comparison among the methods with respect to the L^2 relative errors, that is, $E^{\boldsymbol{u}}_{L^2}(t)$ and $E^{p}_{L^2}(t)$, is not included here for the sake of brevity. The reader can find this analysis in the Supporting Information (Section B.1).

The L^2 velocity and pressure norms are displayed in Figure 8, whereas Figure 9 displays the velocity H^1 seminorms. We recall that the objective functions in algorithms χ-Opt-EFR$^{\boldsymbol{u}}_{\text{global}}$, χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$, and χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ include the information on this specific metric. Hence, we expect better results in these cases with respect to local optimization approaches. As expected, Figures 8 and 9 show that:

- χ-Opt-EFR$^{\boldsymbol{u}}_{\text{local}}$ and χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{local}}$ simulations produce less accurate approximations than in the cases of global optimizations;

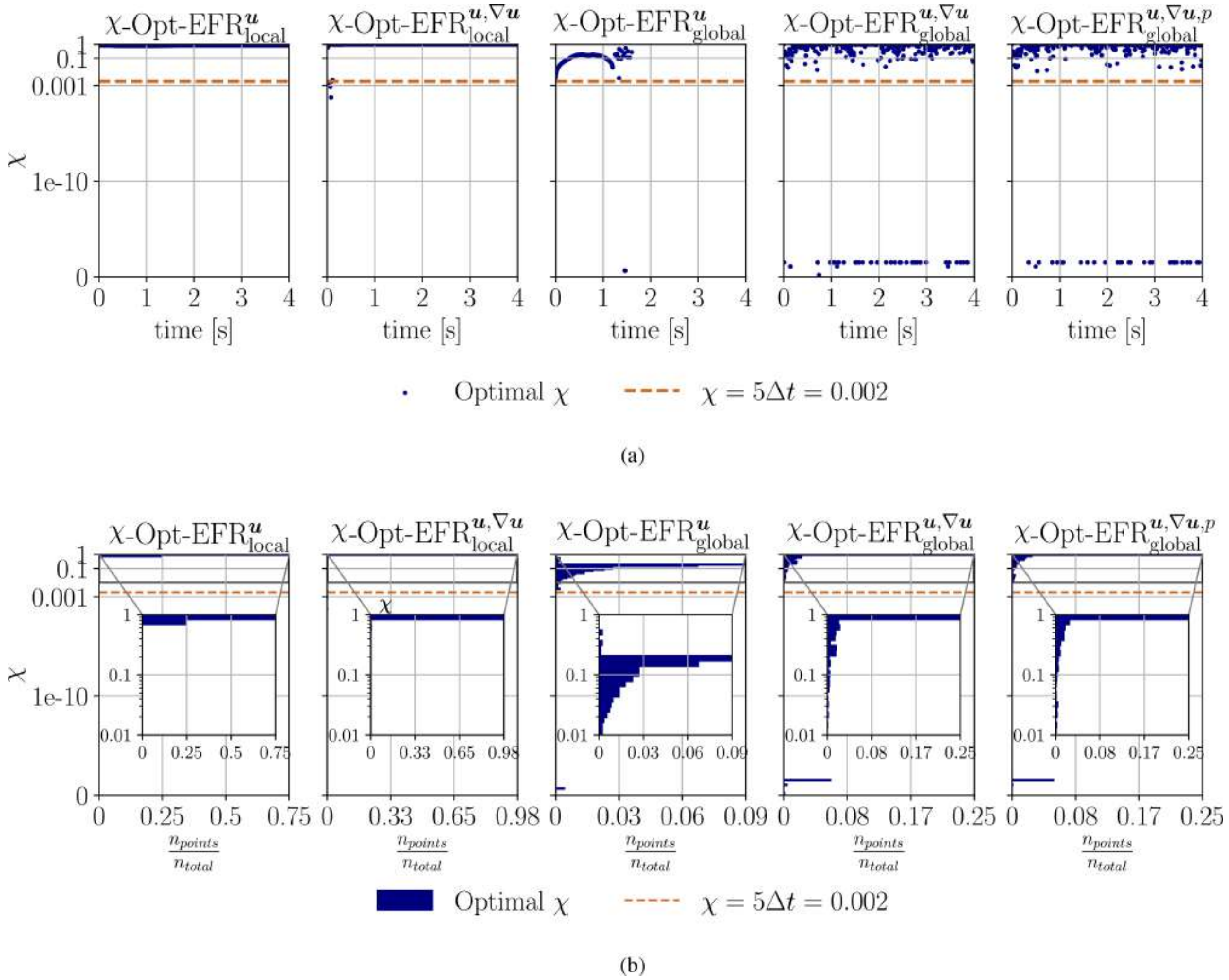


FIGURE 5 | Optimal value of $\chi(t)$ in the χ-Opt-EFR algorithms (A), and count of observations of the parameter values (B). (a) Optimal value of $\chi(t)$, (b) Count of observations of the parameter values.

- χ-Opt-EFR$^{\boldsymbol{u}}_{\text{global}}$ shows a good agreement in the L^2 velocity norm (until $t = 1.6$) with respect to the reference counterpart, because it is exactly the metric optimized in the optimization algorithm, while the pressure norm and the velocity gradients are not accurately predicted. Indeed, this approach yields a poor approximation of the velocity and pressure fields, as can be seen in Figures 6 and 7d;
- Approaches χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ provide the best results, and show a good agreement in the H^1 velocity seminorm and the pressure norm. This is confirmed by the zoomed-in area in Figure 9, which highlights the oscillatory behavior of the Opt-EFR norms. This behavior seems to reproduce the DNS reference norm, while the standard EFR is over-diffusive and, hence, has a constant norm over time;
- We expect that the addition of the pressure contribution in χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ would be beneficial and improve the pressure accuracy with respect to χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$. However, in both cases, the pressure norm has a chaotic trend in time. The graphical comparison between Figure 7e,f shows that the addition of the pressure discrepancy in the objective function only slightly increases the pressure accuracy.

Remark 4. To highlight the reason why the procedure χ-Opt-EFR$^{\boldsymbol{u}}_{\text{global}}$ blows up at $t = 1.6$, in Figure 8 we display the pressure norm, which starts increasing after $t = 1$, until it completely explodes, as can be seen in Figure 7d. This happens because the optimization algorithm for parameter χ does not converge to a global optimum.

4.2 | Numerical Results for δ-Opt-EF

In this section, we present results of the δ-Opt-EF algorithms, namely the ones optimizing parameter $\delta(t)$, for a fixed relaxation parameter, $\chi = 1$. Figure 10a shows the trend of the optimized filter parameter in time.

The results here are similar to the results obtained in Subsection 4.1 with the global optimization of parameter $\chi(t)$, namely $\delta(t)$ is oscillating, but in this case in the prescribed range $[1 \times 10^{-5}, 1 \times 10^{-3}]$. In particular, in the δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ algorithm, the optimal parameter converges to 1×10^{-3} in the first time steps, and then it oscillates but most of the time converges to 1×10^{-5}. Figure 10b presents the distribution of the parameter values, and

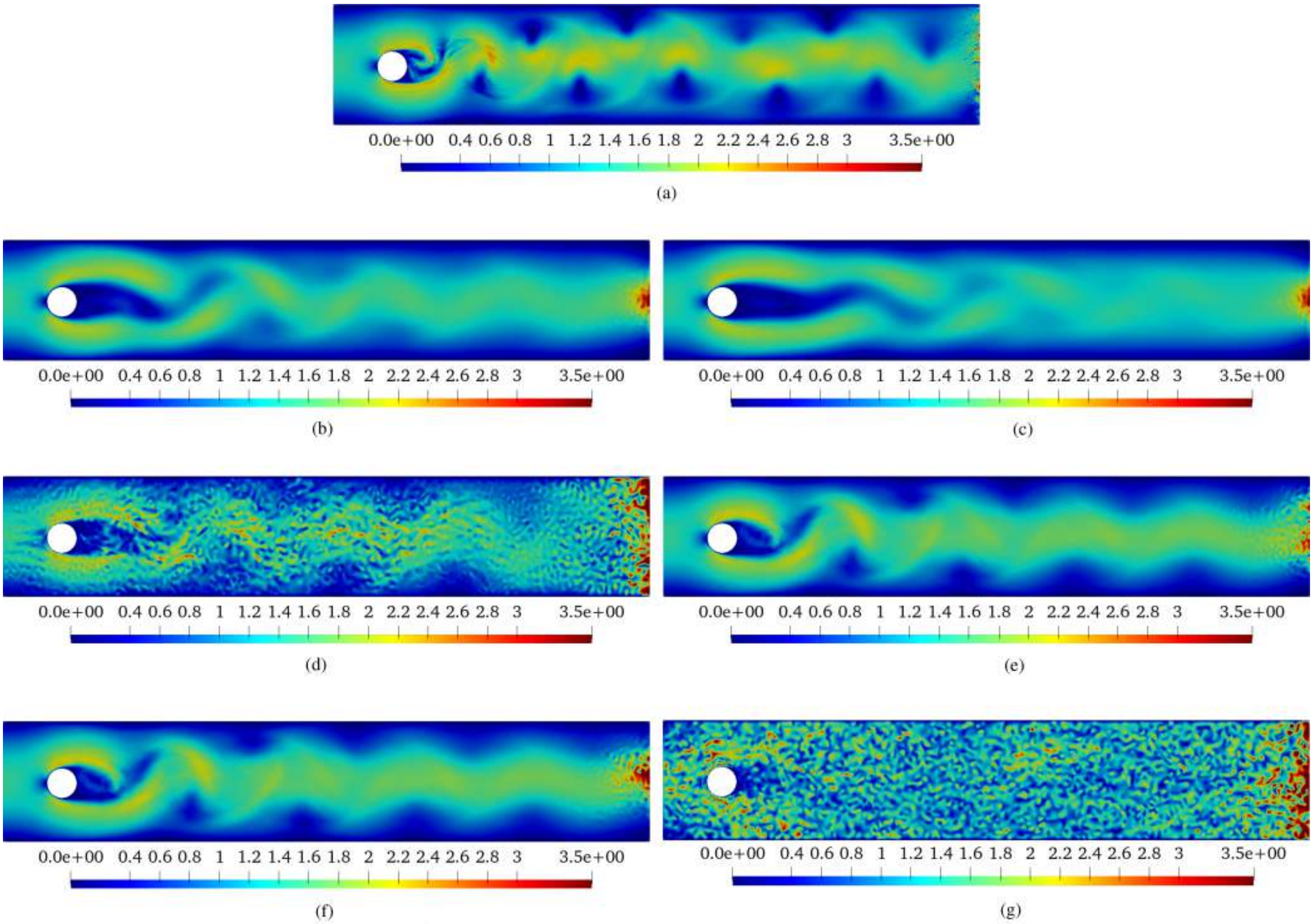


FIGURE 6 | Velocity fields at final time instance $t = 4$ for standard EFR, χ-Opt-EFR results, and reference DNS simulation. (a) DNS-velocity ($t = 4$), (b) χ-Opt-EFR$^{\boldsymbol{u}}_{\text{local}}$-velocity ($t = 4$), (c) χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{local}}$-velocity ($t = 4$), (d) χ-Opt-EFR$^{\boldsymbol{u}}_{\text{global}}$-velocity ($t = 1.6$), (e) χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$-velocity ($t = 4$), (f) χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$-velocity ($t = 4$), and (g) Standard EFR ($\delta = \eta$, $\chi = 5\Delta t$)-velocity ($t = 4$).

it shows that the most frequently occurring value is 1×10^{-5} in all cases. We note that the oscillating behavior between this minimum value and higher values is necessary to alleviate the spurious oscillations appearing in the solution.

The trend of the norms in Figures 11 and 12, and the graphical results in Figures 13 and 14 lead to the following conclusions:

- The optimized algorithms give a better representation of the reference fields, outperforming the over-diffusive approximation of standard EF with fixed δ;
- Comparing the L^2 norms (Figure 11), we can see that the most accurate result is obtained using the δ-Opt-EF$^{\boldsymbol{u}}_{\text{global}}$ algorithm, which directly includes the measured metric inside the objective function. However, this algorithm provides inaccurate results, as seen both from the L^2 pressure norm plots and the H^1 velocity norm plots. This is confirmed by the graphical results (Figures 13 and 14b), which show noisy and poor approximations;
- The δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and the δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ approaches are more accurate than the δ-Opt-EF$^{\boldsymbol{u}}_{\text{global}}$ one, as can be seen from the velocity H^1 and the pressure L^2 norm plots. These approaches are also closer to the reference DNS data than the standard EF, as highlighted in the zoomed-in areas. Also, the graphical results (Figures 13 and 14c,d) confirm a good agreement with the reference DNS data (displayed in Figures 13 and 14a);
- As noticed for the χ-Opt-EFR algorithms, the inclusion of the pressure contribution δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ does not improve the pressure accuracy and negatively impacts the computational cost of the method.

4.3 | Numerical Results for $\delta\chi$–Opt-EFR

In this final part, we discuss the results of the double optimization algorithms $\delta\chi$-Opt-EFR. Since the algorithms χ-Opt-EFR$^{\boldsymbol{u}}_{\text{global}}$ and δ-Opt-EF$^{\boldsymbol{u}}_{\text{global}}$ proved to be inaccurate in the previous Subsections 4.1 and 4.2, we here only consider algorithms $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$.

Figure 15a shows the optimal parameters' trend in time for the two proposed algorithms. The results for parameter $\delta(t)$ are similar in the two cases. However, the relaxation parameter $\chi(t)$ has statistically higher values in algorithm $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$, as can

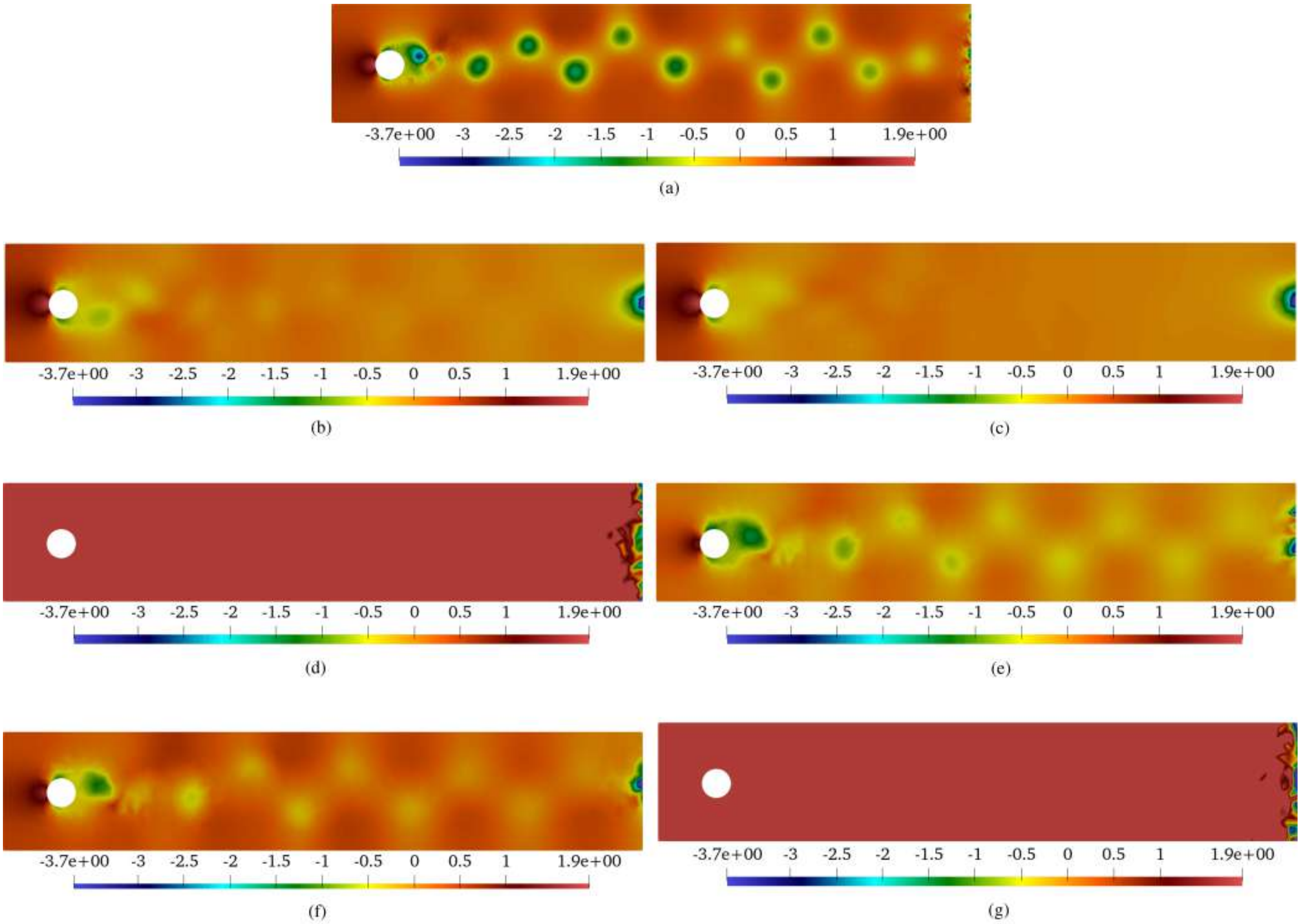


FIGURE 7 | Pressure fields at final time instance $t = 4$ for Standard EFR, χ-Opt-EFR results, and for the reference DNS simulation. (a) DNS-pressure ($t = 4$), (b) χ-Opt-EFR$^{\boldsymbol{u}}_{\text{local}}$-pressure ($t = 4$), (c) χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{local}}$-pressure ($t = 4$), (d) χ-Opt-EFR$^{\boldsymbol{u}}_{\text{global}}$-pressure ($t = 1.6$), (e) χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$-pressure ($t = 4$), (f) χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$-pressure ($t = 4$), and (g) Standard EFR ($\delta = \eta$, $\chi = 5\Delta t$)-pressure ($t = 4$).

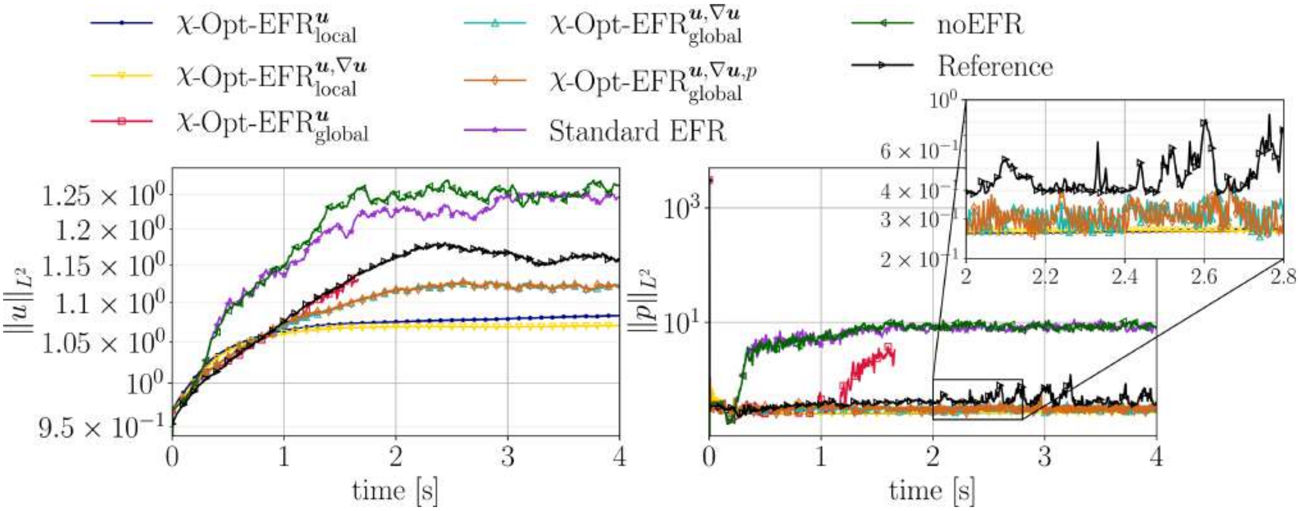


FIGURE 8 | Velocity and pressure norms of χ-Opt-EFR simulations, of the reference solutions projected on the coarse mesh, and of standard EFR. The pressure plot (right) also includes a box with a zoomed-in area in the time interval [2, 2.8].

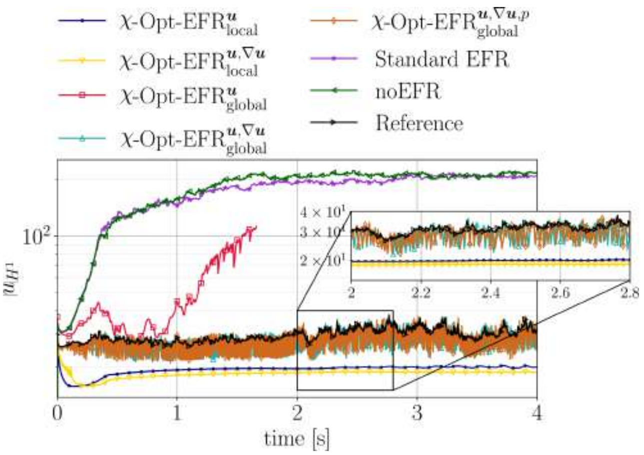


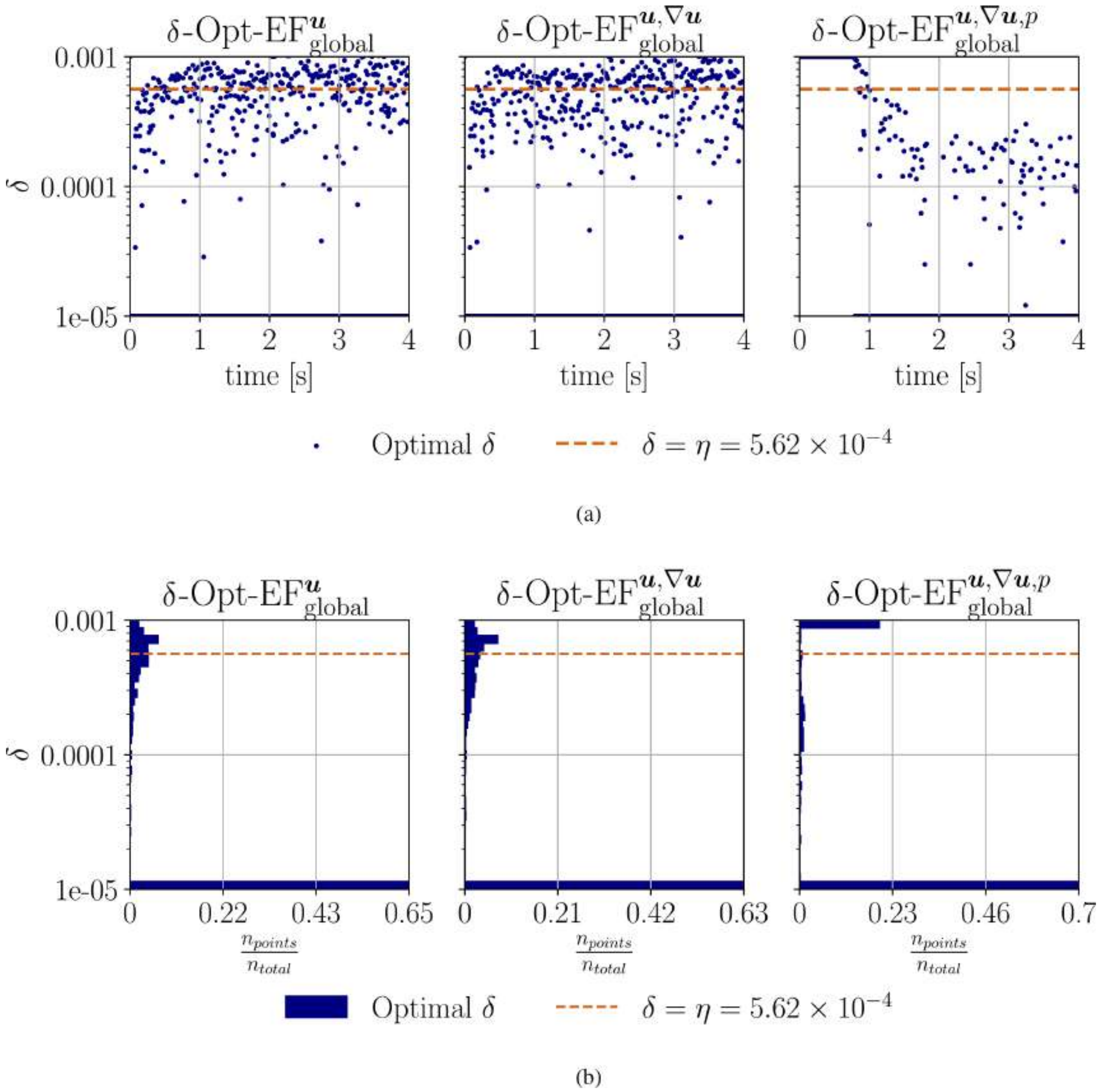
FIGURE 9 | Velocity H^1 norms of χ-Opt-EFR simulations, of the reference solutions, projected on the coarse mesh, and of standard EFR. *Note.* The plot also includes a box with a zoomed-in area in the time range [2, 2.8].

be seen from the histogram in Figure 15b. Moreover, in the double optimizations the optimal $\chi(t)$ is almost always in the range $[1\times10^{-1}, 1]$, while in optimizations χ-Opt-EFR$_{\text{global}}$ (Figure 5) it has a more oscillating behavior, and it also converges to smaller values.

The results for the L^2 and H^1 norms (Figures 16 and 17) are similar for the two algorithms proposed ($\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$), as also highlighted in the zoomed-in areas in the plots. If we also look at the graphical representations of the velocity and pressure fields at $t = 4$ in Figures 18 and 19c,d, the behavior is quite similar. However, the final pressure field is slightly more accurate when the functional also takes into account the pressure contribution, namely in $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$.

4.4 | On the Role of the Optimization Step

The previous analysis was performed with a fixed optimization step, $\Delta t_{opt} = 10$. However, the optimization step plays an



(a)

(b)

FIGURE 10 | Optimal value of $\delta(t)$ in the δ-Opt-EF algorithms (A), and count of observations of the parameter values (B). (a) The first subfigure description, (b) The second subfigure description.

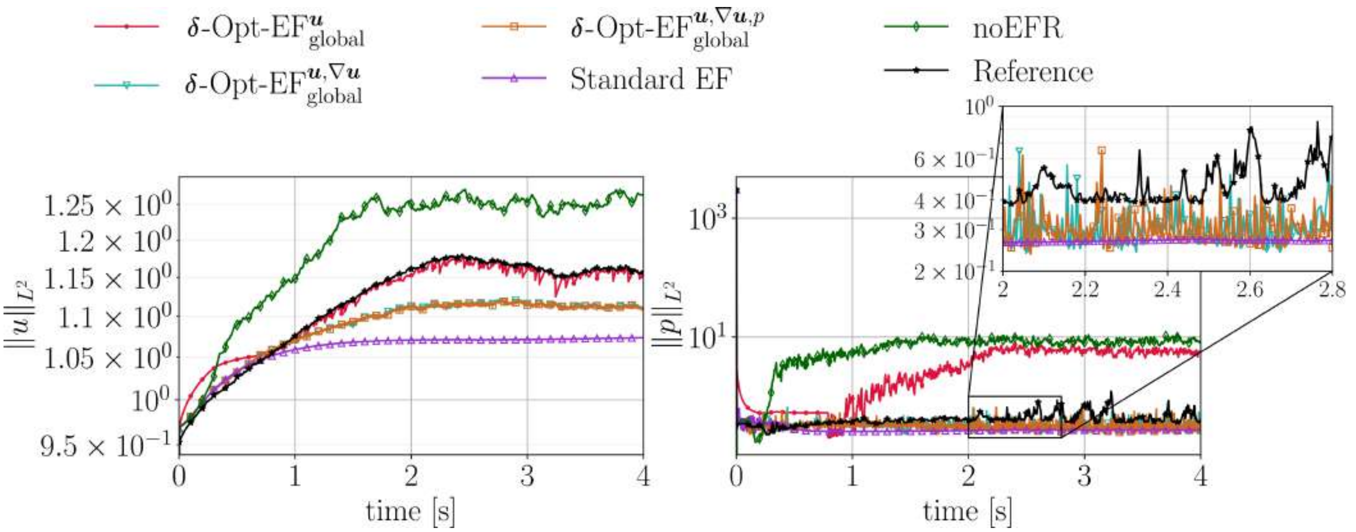


FIGURE 11 | Velocity and pressure norms of δ-Opt-EF simulations, of the EF simulation with fixed δ, and of the reference solutions, projected on the coarse mesh. The pressure plot (right) also includes a box with a zoomed-in area in the time interval $[2, 2.8]$.

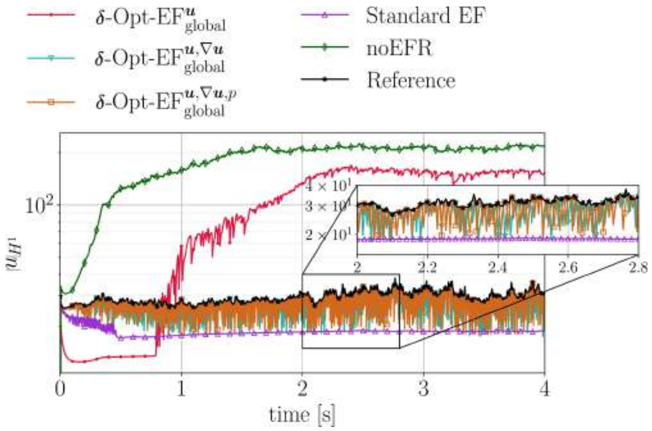


FIGURE 12 | Velocity H^1 norms of δ-Opt-EF simulations, of the EF simulation with fixed δ, and of the reference solutions, projected on the coarse mesh. The plot also includes a box with a zoomed-in area in the time interval $[2, 2.8]$.

important role in ensuring a good compromise between the accuracy and the efficiency of the described approach. Hence, this section is dedicated to the comparison of the different types of optimization, in terms of **efficiency** and **accuracy**. For each algorithm, we also extend the analysis to different optimization steps, namely $\Delta t_{opt} = k\Delta t$, where the positive integer k satisfies $k \in \{10, 20, \ldots, 90, 100\}$.

We measure the **accuracy** using metrics $\overline{\mathcal{L}^{\boldsymbol{u}}_{\text{global}}(\boldsymbol{\mu}^n)}^n$, $\overline{\mathcal{L}^{\nabla\boldsymbol{u}}_{\text{global}}(\boldsymbol{\mu}^n)}^n$, and $\overline{\mathcal{L}^{p}_{\text{global}}(\boldsymbol{\mu}^n)}^n$, where the operator $\overline{(\cdot)}^n$ is a time average, and the terms $\mathcal{L}^{(\cdot)}_{\text{global}}(\boldsymbol{\mu}^n)$ are the global contributions in the optimization introduced in Equation (6).

We measure the **efficiency** in terms of the wall-clock time, expressed in relative percentage with respect to the DNS time (bottom x-axis) and in absolute time (top x-axis).

The results of the $\delta\chi$-Opt-EFR simulations with different k values are presented in the Pareto plots in Figures 20 and 21, where the k value is visualized in the boxes next to the corresponding markers.

Since in the plots the x-axis represents the computational time and the y-axis represents the errors, we expect the best-performing simulations to be located in the lower left part of the plots, which corresponds to low errors and low computational time. Moreover, our goal is to discard simulations taking place in the upper left part, corresponding to high errors and low computational cost, and in the lower right part, corresponding to low errors and high computational cost.

Figures 20 and 21 yield the following conclusions:

- The standard EFR and the noEFR simulations lead to higher errors than the optimized-EFR approaches, as we expect. In particular, for the pressure and the velocity gradients, $\delta\chi$-Opt-EFR methods are able to improve the accuracy by **two** orders of magnitude. Moreover, the standard EFR and the noEFR simulations also need more computational time than some of the $\delta\chi$-Opt-EFR simulations. This may be due to the fact that the noEFR and standard-EFR simulations are **unstable and noisy**, and, hence, the FEM Newton solver needs more iterations to converge to a solution.
- With respect to the relative wall-clock time, all the optimized-EFR simulations have **high gain in time** with respect to the DNS simulation. However, the absolute time suggests that the range of all the $\delta\chi$-Opt-EFR simulations is $[4, 9]$ hours. This means that varying the optimization time window, the change in the computational time of the $\delta\chi$-Opt-EFR simulations may be on the order of hours.
- As expected, the simulations with a smaller optimization step, that is, $10\Delta t$ or $20\Delta t$, need more time than the simulations with higher optimization steps, but are more accurate. We also note that the accuracy among different

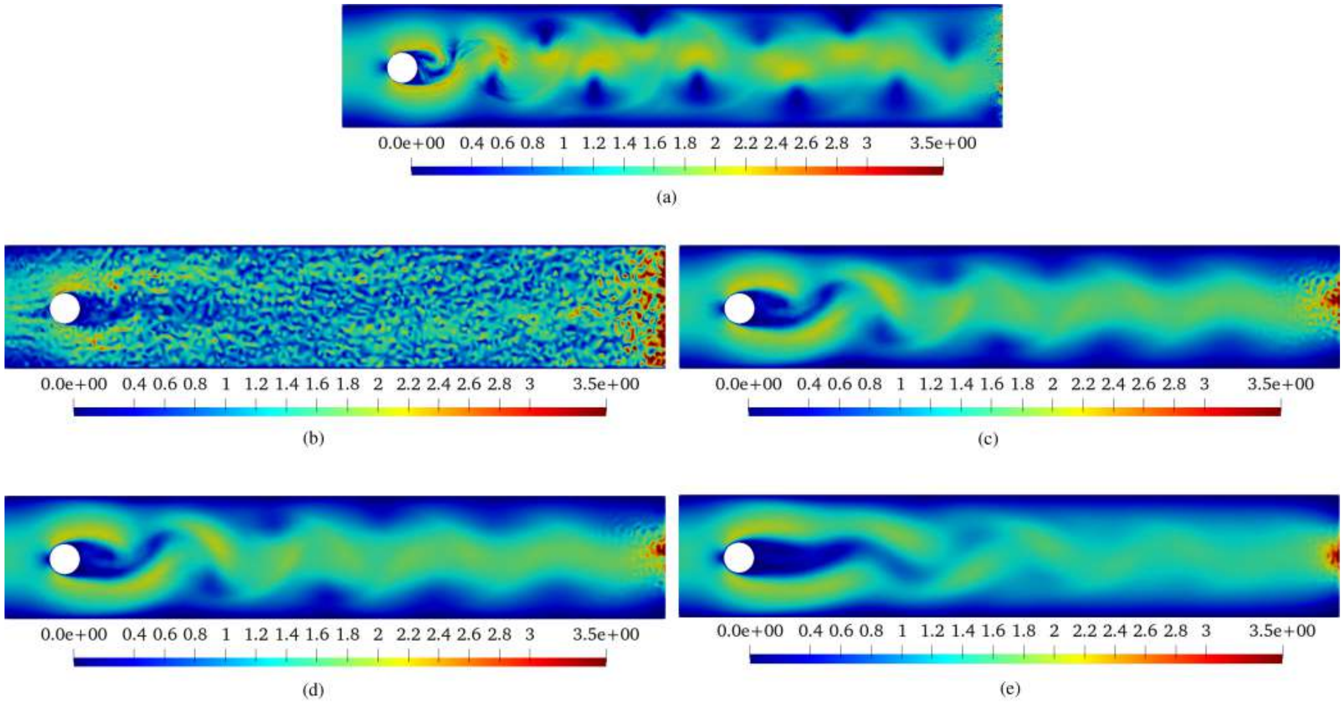


FIGURE 13 | Velocity fields at final time $t = 4$ for standard EF, δ-Opt-EF, and reference DNS. (a) DNS-velocity ($t = 4$), (b) δ-Opt-EF$^{\boldsymbol{u}}_{\text{global}}$-velocity ($t = 4$), (c) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$-velocity ($t = 4$), (d) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$-velocity ($t = 4$), and (e) Standard EF ($\delta = \eta$)-velocity ($t = 4$).

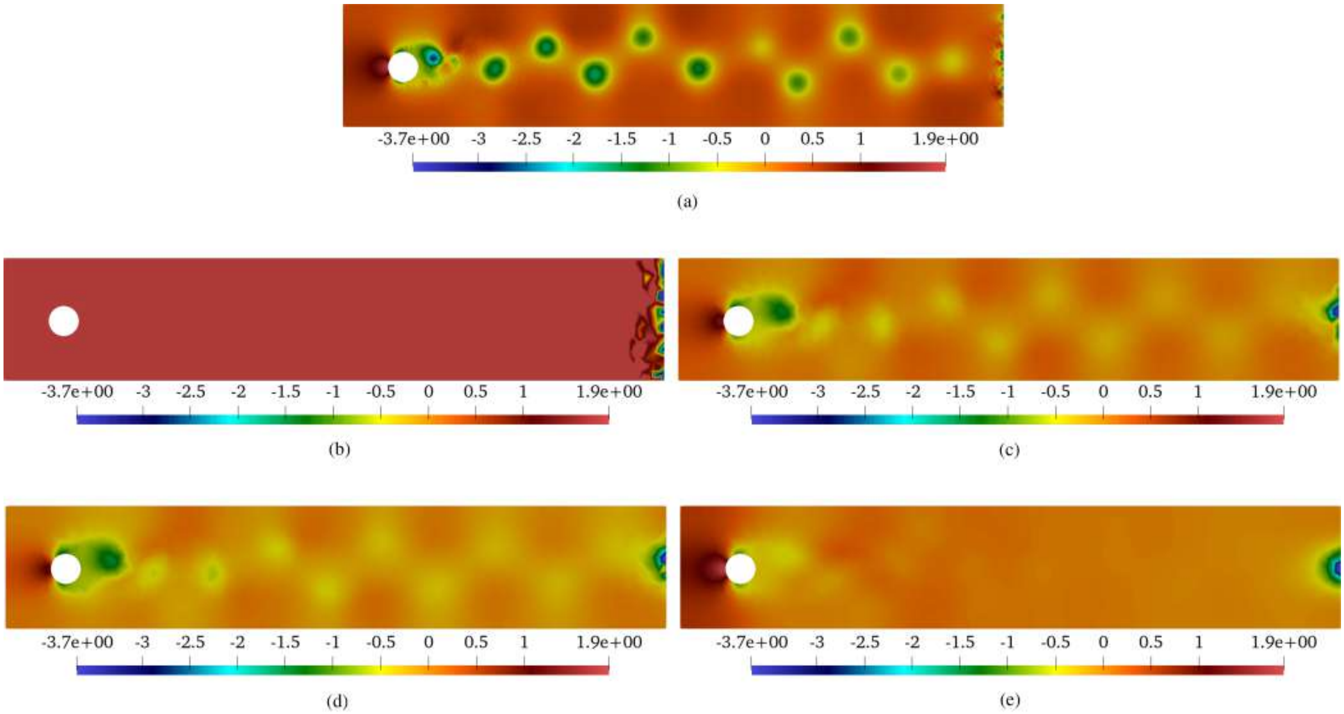


FIGURE 14 | Pressure fields at final time $t = 4$ for standard EF, δ-Opt-EF, and reference DNS. (a) DNS-pressure ($t = 4$), (b) δ-Opt-EF$^{\boldsymbol{u}}_{\text{global}}$-pressure ($t = 4$), (c) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$-pressure ($t = 4$), (d) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$-pressure ($t = 4$), and (e) Standard EF ($\delta = \eta$)-pressure ($t = 4$).

optimization steps is still comparable in terms of the L^2 norms, but not in terms of the H^1 velocity seminorm. This may be due to the fact that the velocity gradients get too far from the global optimum before a novel optimization is performed. Hence, the following optimization step is no longer able to converge and to "fix" the spatial gradients.

- The $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ strategy slightly improves pressure accuracy. However, it is more time-consuming than the

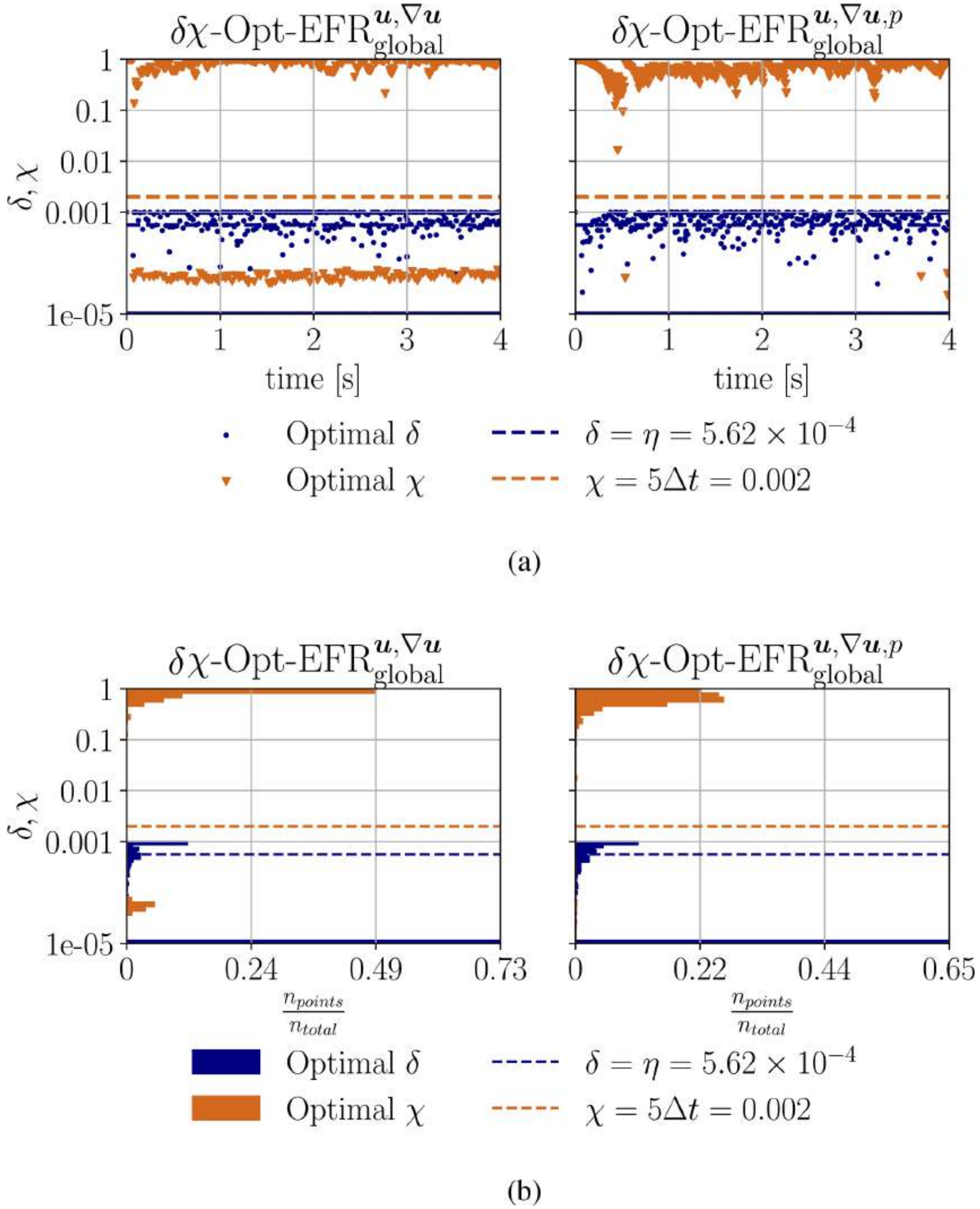


FIGURE 15 | Optimal value of $\delta(t)$ and $\chi(t)$ in the $\delta\chi$-Opt-EFR algorithms (A), and relative parameter distribution (B).

$\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ strategy, without yielding a significant gain in accuracy.

The same sensitivity study for χ-Opt-EFR and δ-Opt-EF can be found in the supplementary results (Sections B.1.1 and B.2.2). The above considerations for $\delta\chi$-Opt-EFR hold true also for the other two strategies.

Remark 5. The Pareto plots in Figures 20 and 21 show how the optimization step influences the simulation wall-clock time. One would expect that simulations with larger time windows are faster than simulations with smaller time windows. However, it is important to underline that this general statement is not always satisfied. For example, simulations with $\Delta t_{opt} \in \{70, 80, 90, 100\}$ take more time than simulations with $\Delta t_{opt} \in \{40, 50, 60\}$. This is because larger optimization steps lead to more unstable systems (i.e., spurious oscillations and presence of noise), and in this scenario, the Newton method needs more iterations to converge to the solution.

4.5 | Final Discussion

We here consider a comparison among the algorithms having the best performance in the previous sections, namely $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$.

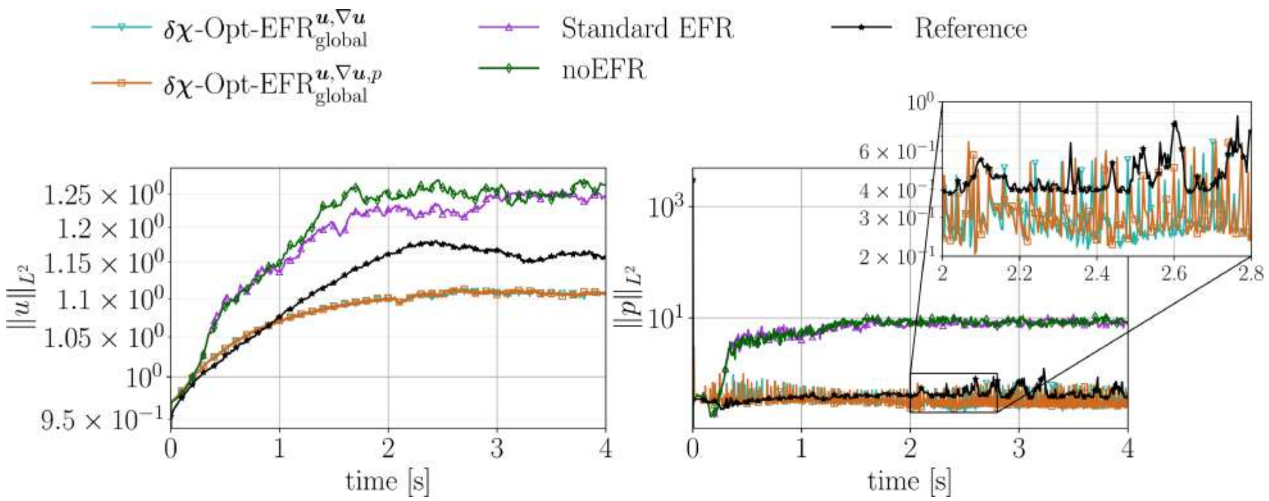


FIGURE 16 | Velocity and pressure norms of $\delta\chi$-Opt-EFR simulations and reference solutions projected on the coarse mesh. The pressure plot (right) also includes a box with a zoomed-in area in the time interval $[2, 2.8]$.

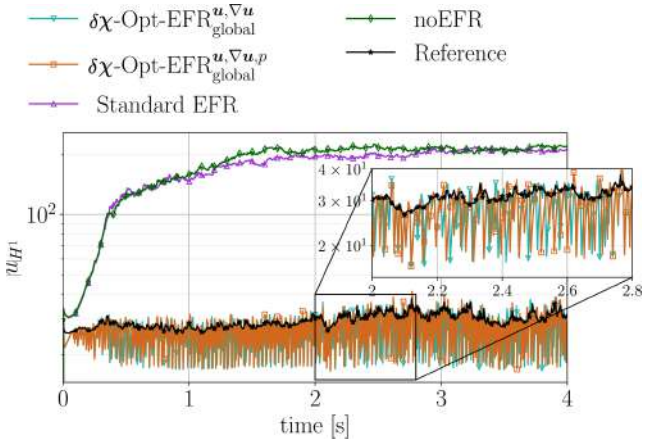


FIGURE 17 | Velocity H^1 norms of $\delta\chi$-Opt-EFR simulations and reference solutions projected on the coarse mesh. The plot also includes a box with a zoomed-in area in the time interval $[2, 2.8]$.

The comparison in the norms in Figures 22 and 23 shows that all the results are similar. However, in the H^1 velocity norm, algorithm χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ is less oscillating than the others (as one can notice from the zoomed-in areas), and is also slightly more accurate in the L^2 velocity norm.

In general, there is a consistent enhancement in the accuracy between the standard and the Opt-EFR strategies. This improvement is accompanied by slightly longer times in the case of Algorithm $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$, but considerably longer times for strategy $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$, since it includes an evolve step within each optimization pass. $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ slightly improves the accuracy obtained with $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ in terms of velocity gradients and pressure discrepancy.

In addition, in what follows we report a series of considerations concerning the comparison among all the algorithms proposed in Sections 4.1, 4.2, and 4.3.

- The χ-Opt-EFR strategies optimizing *local* quantities, namely algorithms χ-Opt-EFR$^{\boldsymbol{u}}_{\text{local}}$ and χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{local}}$, provide over-diffusive results and are not able to fully capture the dynamics of the test case. For this reason, in Sections 4.2 and 4.3, we only consider the strategies with *global* loss contributions.
- The strategies of type $(\cdot)^{\boldsymbol{u}}_{\text{global}}$, optimizing the discrepancy on the *global* quantity $\|\boldsymbol{u}\|_{L^2(\Omega)}$, lead to poor results. In particular, the optimized velocity field is characterized by spurious oscillations. That motivates the addition of the gradient contribution to the loss functional.
- The strategies of type $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$, which include the discrepancy in both $\|\boldsymbol{u}\|_{L^2(\Omega)}$ and $\|\nabla\boldsymbol{u}\|_{L^2(\Omega)}$, provide the best results in terms of velocity accuracy. Therefore, the inclusion of the gradient contribution within the optimization process is indispensable for achieving favorable outcomes in velocity accuracy.
- The strategies considering the pressure contribution in the loss functional, namely $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$, yield similar results to the algorithms $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ in terms of velocity accuracy. However, the inclusion of the pressure discrepancy in the loss functional leads to better results in terms of pressure accuracy.
- As discussed previously, the computational times of Opt-EFR algorithms are comparable to the standard EFR in strategy $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$, but with considerably better results. However, the addition of the pressure contribution in the loss functional leads to longer computational times, without a consistent improvement in the accuracy. Moreover, in all the Opt-EFR algorithms, there is a consistent gain in time with respect to the DNS simulation.
- We can notice that all the algorithms of type $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ outperforms the results of standard EFR ($\delta = \eta$, $\chi = 5\Delta t$). Moreover, δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ with $\Delta t_{opt} = 10\Delta t$ or $\Delta t_{opt} = 20\Delta t$ outperform the results of standard EF ($\delta = \eta$). Indeed, standard and fixed values for the

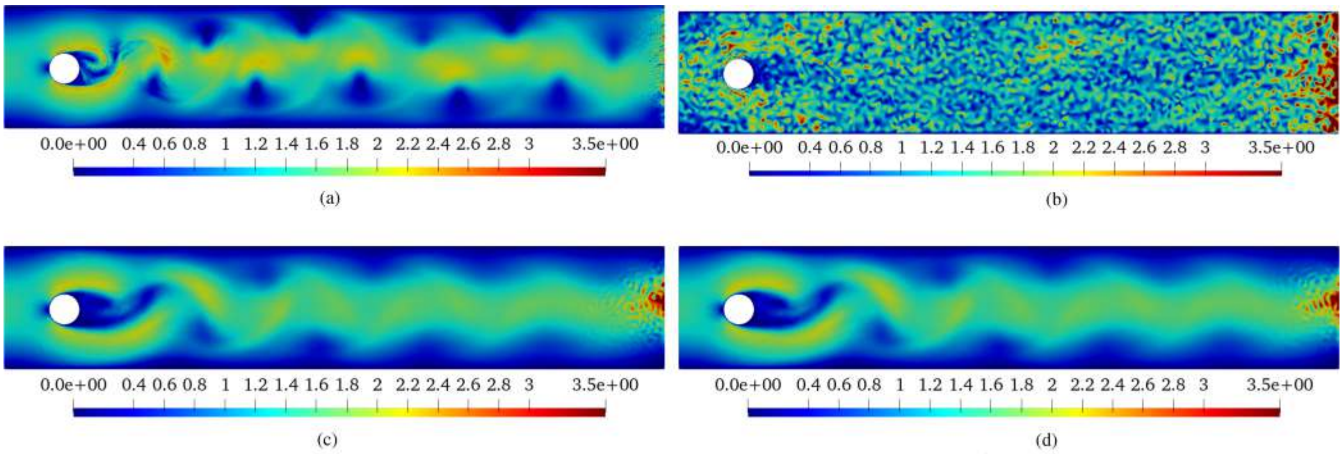


FIGURE 18 | Velocity fields at final $t = 4$ for $\delta\chi$-Opt-EFR simulations, DNS reference, and standard EFR. (a) DNS-velocity ($t = 4$), (b) Standard EFR-velocity ($t = 4$), (c) $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$-velocity ($t = 4$), and (d) $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$-velocity ($t = 4$).

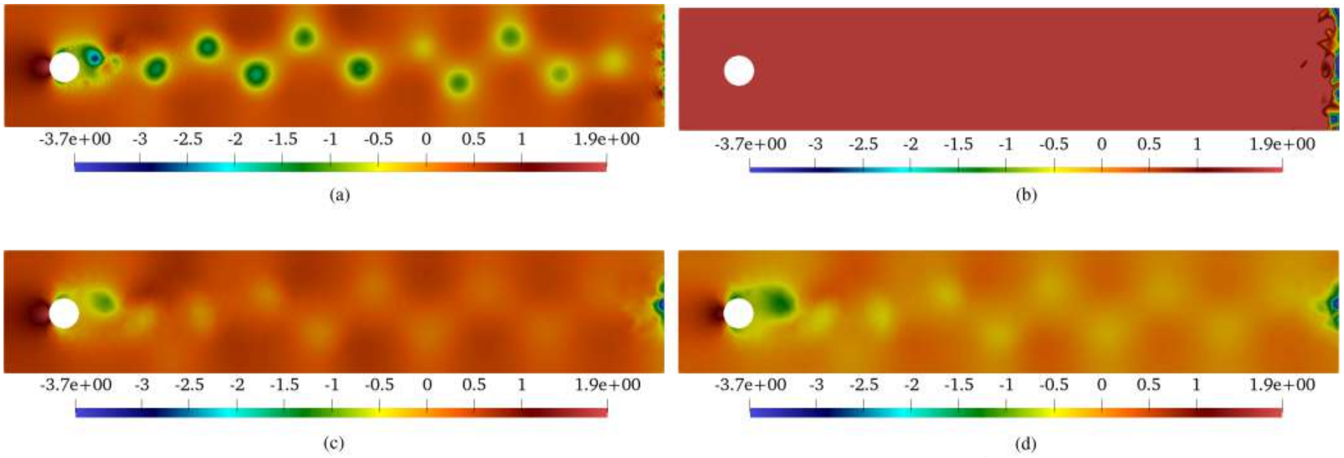


FIGURE 19 | Pressure fields at final time $t = 4$ for $\delta\chi$-Opt-EFR, DNS reference, and standard EFR. (a) DNS-pressure ($t = 4$), (b) Standard EFR-pressure ($t = 4$), (c) $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$-pressure ($t = 4$), and (d) $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$-pressure ($t = 4$).

filter and the relaxation parameter lead either to spurious oscillations or to over-diffusive fields. Moreover, the computational time of standard EFR is higher than that of the optimized methods because the simulation is not stable, and the Newton method needs more iterations to converge to a solution. This suggests that the optimized strategies also work as numerical stabilizations.

- As a general observation, the velocity H^1 seminorm has an oscillatory behavior in all the examined cases. This is linked to a similar chaotic behavior of the optimal parameter values. Indeed, the optimization algorithm minimizes the discrepancy in the velocity gradients' norm, so:
  - If the norm is far from the reference one, that is, the velocity starts to exhibit a noisy and nonphysical behavior, which means that the parameters are too small and the algorithm converges to large optimal values;
  - Consequently, after Δt_{opt}, the norm is closer to the reference DNS, and the simulation is more stable. Hence, the simulation no longer requires large filter radii and/or relaxation parameters, and the algorithm converges to smaller optimal values.

  This mechanism repeats as the simulation evolves, leading to oscillatory behaviors in both parameters and norms.

As an overview of the results, we include two additional Figures:

- Figure 24 represents the time-average (orange lines) and confidence interval (boxes) of the relative errors in the velocity H^1 norm and pressure L^2 norm. It confirms a significant improvement in the Opt-EFR strategies with respect to the baseline EFR/EF.
- Figure 25 displays the gain increase in the accuracy of Opt-EFR results with respect to the baselines (standard EFR/EF). We measure the "*gain*" metric as the time-averaged relative error of the Opt-EFR solutions with respect to standard EFR and standard EF results. We conclude that Opt-EFR strategies significantly enhance the

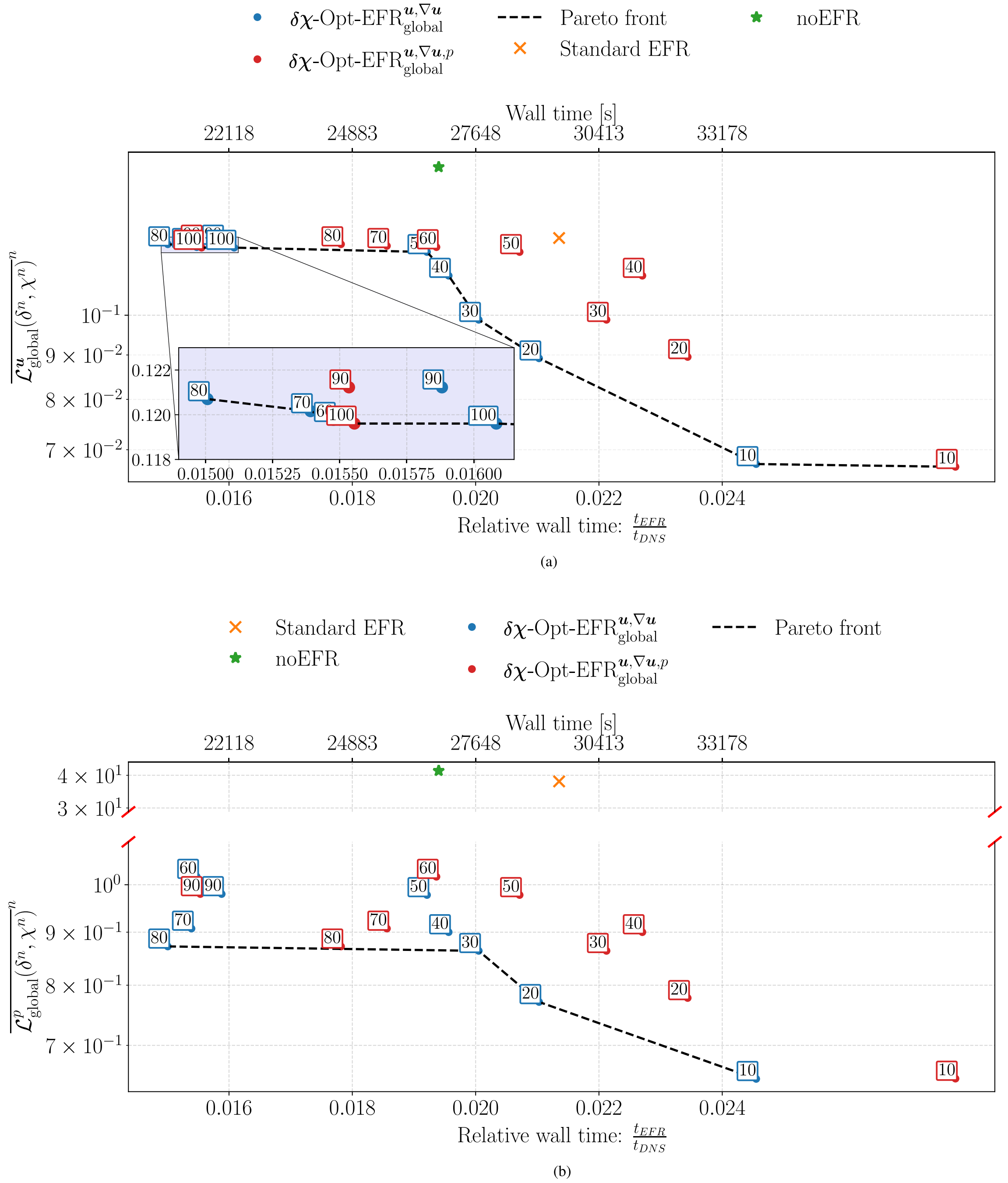


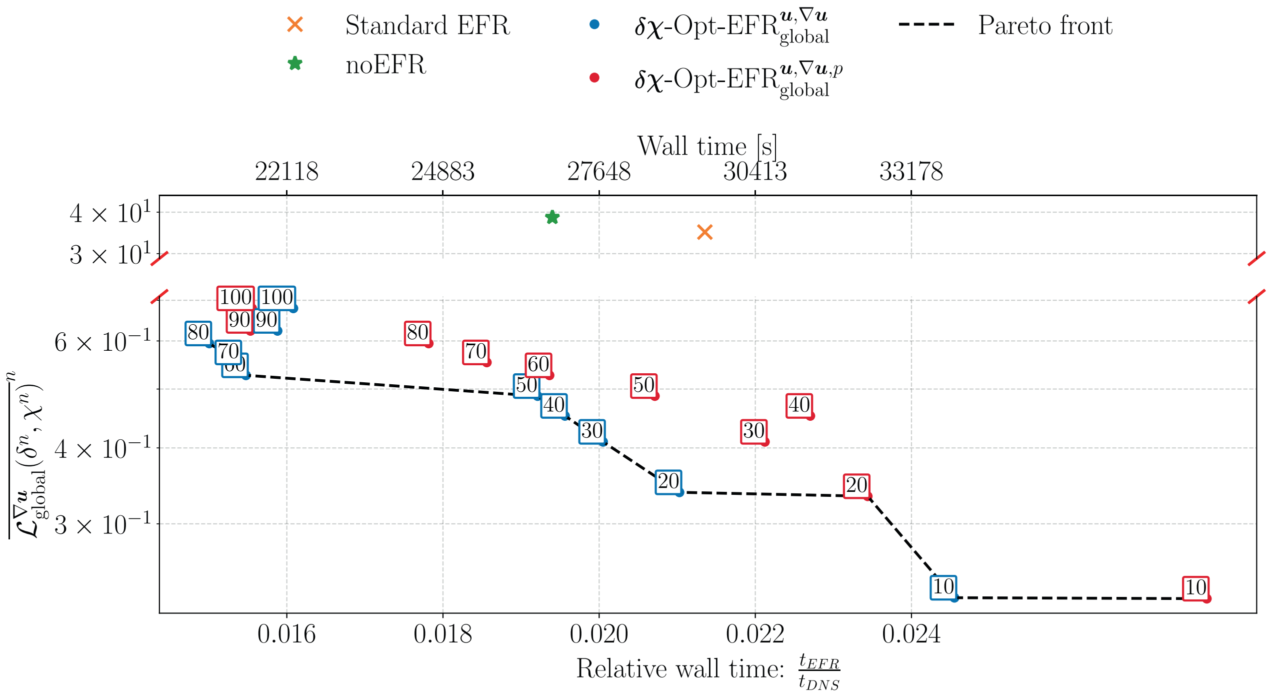
FIGURE 20 | Representation of how the optimization step affects the efficiency and the accuracy of the final velocity or pressure approximation in $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ simulations. The efficiency, on the x-axis, is measured through the relative wall-clock time with respect to the DNS time. The accuracy, on the y-axis, is measured with the L^2 velocity and pressure norms. (a) Discrepancy on the velocity L^2 norm, (b) Discrepancy on the pressure L^2 norm.

accuracy of standard methods. In particular, we reach an average gain of **99** % with respect to standard EFR, and of **83** % (velocity) and **27–30** % (pressure) with respect to standard EF.

Remark 6. Algorithm 1 can be adapted to various test cases and different discretization types, such as finite element and finite volume methods. As expected, the related results will be problem-dependent. We remark, however, that this paper aims to provide a data-driven alternative to the standard EFR technique that enhances the simulation accuracy while maintaining a similar computational cost.

5 | Conclusions and Outlook

In this work, we make a significant contribution to the EFR regularization approach by proposing a **novel data-driven automated tuning of the filter and relaxation parameters**, δ and χ, respectively.



(a) Discrepancy on the velocity H^1 semi-norm

FIGURE 21 | Representation of how the optimization step affects the efficiency and the accuracy of the final velocity or pressure approximation in χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ simulations. The efficiency, on the x-axis, is measured through the relative wall-clock time with respect to the DNS time. The accuracy, on the y-axis, is measured with the H^1 velocity seminorm. (a) Discrepancy on the velocity H^1 semi-norm.

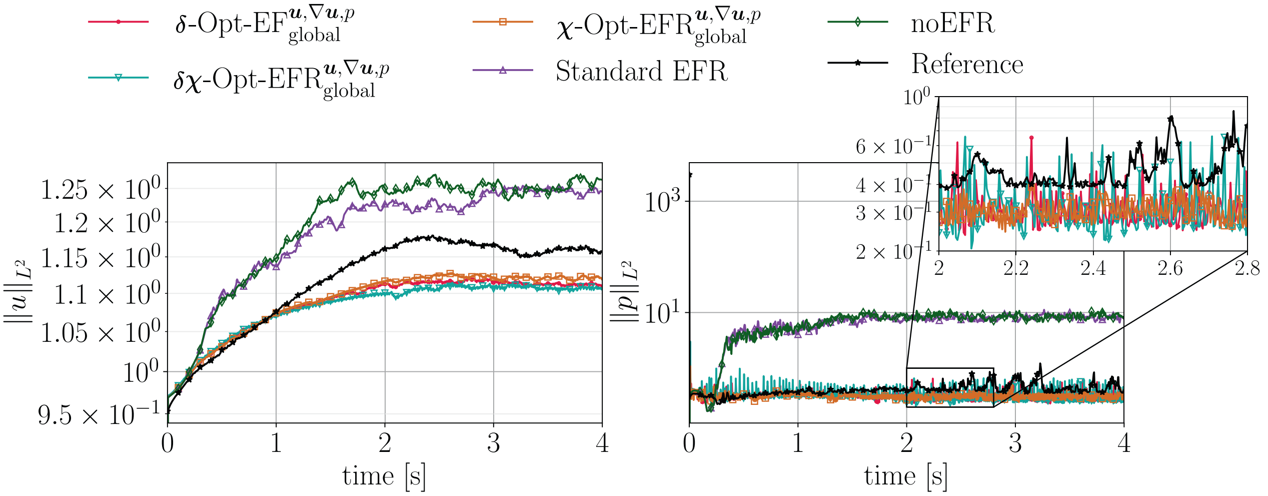


FIGURE 22 | Velocity and pressure norms for algorithms $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ and reference solutions projected on the coarse mesh. The pressure plot (right) also includes a box with a zoomed-in area in the time interval $[2, 2.8]$.

We note that the literature provides guidelines for the values of these parameters. However, existing approaches typically rely on: (i) constant parameter values; and (ii) heuristic choices or problem-dependent tuning, lacking a systematic framework for optimal selection. Our approach introduces an optimization-based methodology (the **Opt-EFR**) to dynamically adapt these parameters over time, improving the robustness and accuracy of EFR simulations. This novelty is especially relevant for under-resolved flow regimes, where fixed parameters often lead to suboptimal results.

In particular, we propose and investigate three novel data-driven optimization algorithms:

- χ-Opt-EFR, optimizing χ for fixed δ;
- δ-Opt-EF, optimizing δ for fixed $\chi = 1$;
- $\delta\chi$-Opt-EFR, optimizing both δ and χ.

In each case, we investigate different objective functions, minimizing the discrepancy between the outcome of the EFR

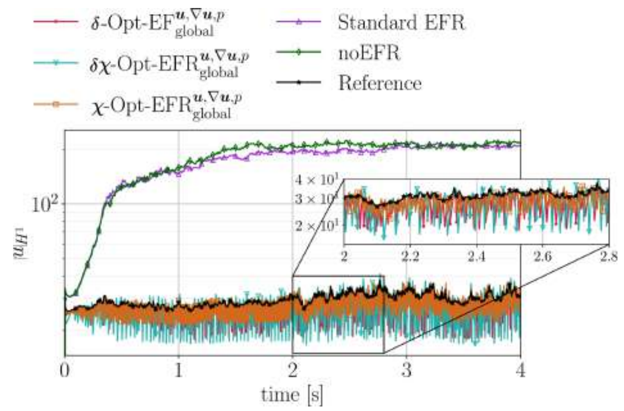


FIGURE 23 | Velocity H^1 norms for algorithms $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ and reference solutions projected on the coarse mesh. The plot also includes a box with a zoomed-in area in the time interval [2, 2.8].

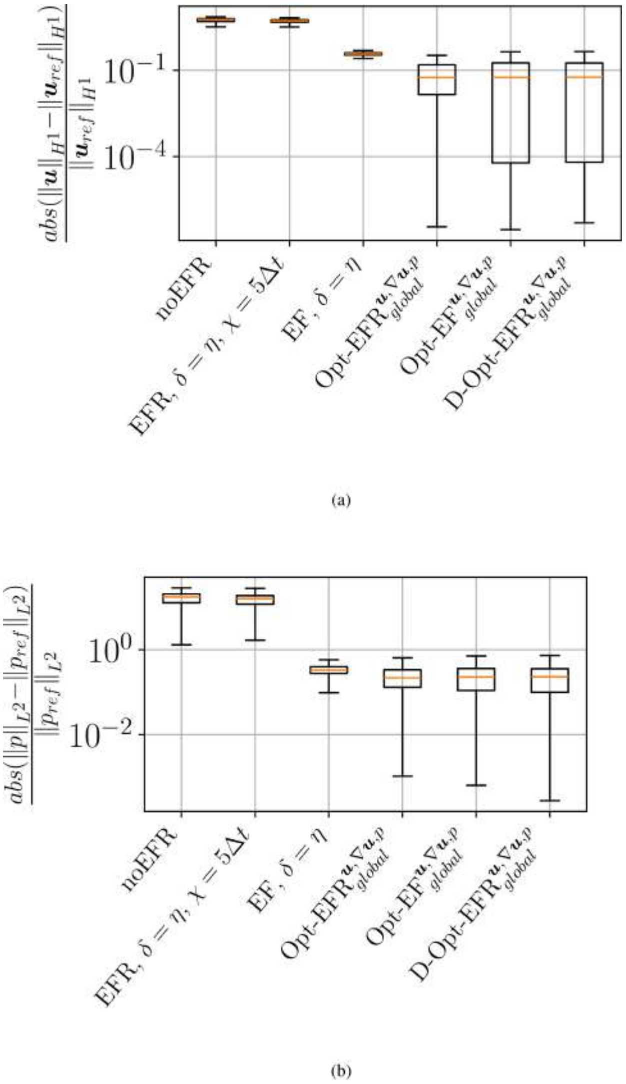


FIGURE 24 | Statistics (median and confidence intervals) for noEFR, standard EFR/EF, and the methods of type $(\cdot)^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$. The results display the relative errors with respect to the DNS reference, and the plots show the time-averaged error (orange line) and the confidence interval (boxes). Specifically, Figure (a) displays the velocity H^1 norm discrepancy, while Figure (b) shows the pressure L^2 norm results. The statistics are computed within the time window [0, 4].

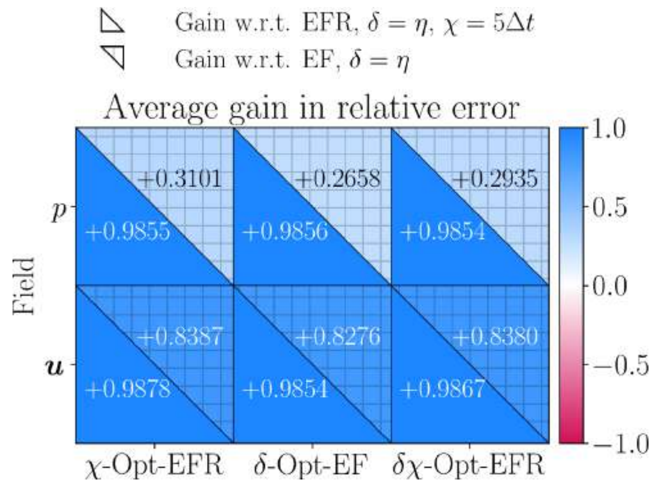


FIGURE 25 | Gain in the relative errors in the H^1 velocity norm and L^2 pressure norm. The results show the time average gain of the best Opt-EFR methods with respect to the standard EFR and standard EF simulations.

simulations (in an under-resolved regime) and a reference DNS (i.e., noEFR performed on a resolved grid).

We perform the numerical simulations on the test case of incompressible unsteady flow past a cylinder at a fixed Reynolds number $Re = 1,000$.

The key new findings of the proposed study are:

1. The optimized-EFR simulations always provide better results than noEFR and standard EFR with fixed parameters.
2. The best results are achieved by including in the objective function the discrepancies of the velocity **gradients** with respect to the reference simulation.
3. The **global objective functions** always yield more accurate results than the corresponding local objective functions.
4. The Opt-EFR simulations have a **similar or smaller computational time** than the standard EFR, even if they include an additional optimization step. This happens because the choice of optimal parameters improves the **numerical stability** of the system and allows for faster convergence.

Despite these promising results, we note as a limitation the **reliance on data** for the optimization, which requires pre-computed DNS data to evaluate the discrepancy between the optimized EFR and the DNS reference. Additionally, the approach is **problem-dependent**, as the algorithm needs to be re-executed in different flow configurations.

As a possible future research direction, the authors would like to consider the application of machine learning approaches, such as **reinforcement learning**, to perform the training phase of the new optimization algorithms separately from the simulation. In this way, the training is performed only once and employs the data from a specific time window, allowing the prediction of the solution in a time extrapolation setting. This approach could significantly reduce the computational overhead and enable real-time parameter adaptation.

Another possible outlook for the project is to extend the new optimization strategies to the framework of reduced-order models by comparing the optimal full-order model parameters with the optimal reduced-order model parameters.

We also highlight the **flexibility** of the Opt-EFR algorithms, which can be trivially extended to other flow scenarios, such as turbulent channel flows or aerodynamic simulations, where adaptive regularization plays a crucial role in improving prediction accuracy.

Author Contributions

Conceptualization: Anna Ivagnes, Maria Strazzullo, Traian Iliescu; Methodology: Anna Ivagnes, Maria Strazzullo, Traian Iliescu; Formal analysis and investigation: Anna Ivagnes; Writing – original draft preparation: Anna Ivagnes, Maria Strazzullo, Traian Iliescu; Writing – review and editing: Maria Strazzullo, Michele Girfoglio, Traian Iliescu; Funding acquisition: Gianluigi Rozza; Supervision: Traian Iliescu, Gianluigi Rozza.

Acknowledgments

This work was partially funded by INdAM-GNCS: Istituto Nazionale di Alta Matematica – Gruppo Nazionale di Calcolo Scientifico, and by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Actions, grant agreement 872442 (ARIA). M. S. and G. R. thank the "20227K44ME-Full and Reduced order modeling of coupled systems: Focus on non-matching methods and automatic learning (FaReX)" project-funded by European Union-Next Generation EU within the PRIN 2022 program (D.D. 104-02/02/2022 Ministero dell'Università e della Ricerca). G. R. acknowledges the consortium iNEST (Interconnected North-East Innovation Ecosystem), Piano Nazionale di Ripresa e Resilienza (PNRR)-Missione 4 Componente 2, Investimento 1.5-D.D. 1058 23/06/2022, ECS00000043, supported by the European Union's NextGenerationEU program. M. S. thanks the ECCOMAS EYIC Grant "CRAFT: Control and Reg-reduction in Applications for Flow Turbulence". M. G. thanks the PON "Research and Innovation on Green-Related Issues" FSE REACT-EU 2021 project. T. I. was funded by the National Science Foundation through grant DMS-2012253. This manuscript reflects only the authors' views and opinions, and the Ministry cannot be considered responsible for them. Open access publishing facilitated by Politecnico di Torino, as part of the Wiley - CRUI-CARE agreement.

Disclosure

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

1. L. C. Berselli, T. Iliescu, and W. J. Layton, *Mathematics of Large Eddy Simulation of Turbulent Flows*, vol. 2 (Springer, 2006).

2. W. J. Layton and L. G. Rebholz, *Approximate Deconvolution Models of Turbulence: Analysis, Phenomenology and Numerical Analysis*, vol. 2042 (Springer, 2012).

3. S. Choi, Y. Kim, and J. H. Hwang, "On the Performance of Generative Adversarial Network for Generating Turbulent Round Jet Flow," *Environmental Fluid Mechanics* 25, no. 1 (2025): 1–19.

4. Y. Li and T. Okaze, "Optimizing Non-Isothermal Turbulent Inflow Generation Using Precursor Method: A Sensitivity Study," *Building and Environment* 269 (2025): 112465.

5. T. C. Rebollo and R. Lewandowski, *Mathematical and Numerical Foundations of Turbulence Models and Applications* (Springer, 2014).

6. S. M. Ahmadi, S. Balahang, and S. Abolfathi, "Predicting the Hydraulic Response of Critical Transport Infrastructures During Extreme Flood Events," *Engineering Applications of Artificial Intelligence* 133 (2024): 108573.

7. M. Guo, D. Goodarzi, J. Pearson, A. Mohammadian, and S. Abolfathi, "The Effects of Permeable Baffles on Hydraulic and Treatment Performance in Retention Ponds," *Journal of Water Process Engineering* 69 (2025): 106593.

8. B. Martínez, M. Guerra, N. Riviere, E. Mignot, and O. Link, "Rans Simulation of Supercritical Open Channel Flows Around Obstacles," *Journal of Hydraulic Research* 63, no. 1 (2025): 1–14.

9. P. Zou, S. Cao, and D. Cao, "Study of the Applicability of Different Turbulence Models in Simulating Vortex-Induced Vibration of a Rectangular Cylinder," *Journal of Wind Engineering and Industrial Aerodynamics* 258 (2025): 106031.

10. J. P. Boyd, *Chebyshev and Fourier Spectral Methods* (Courier Corporation, 2001).

11. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, "Some Algorithms for Unsteady Navier-Stokes Equations," in *Spectral Methods in Fluid Dynamics* (Springer, 1988), 201–239.

12. V. J. Ervin, W. J. Layton, and M. Neda, "Numerical Analysis of Filter-Based Stabilization for Evolution Equations," *SIAM Journal on Numerical Analysis* 50, no. 5 (2012): 2307–2335.

13. P. Fischer, J. Kruse, J. Mullen, H. Tufo, J. Lottes, and S. Kerkemeier, *Nek5000–Open Source Spectral Element CFD Solver* (Argonne National Laboratory, Mathematics and Computer Science Division, 2008).

14. P. Fischer and J. Mullen, "Filter-Based Stabilization of Spectral Element Methods," *Comptes Rendus de L'académie Des Sciences-Series I-Mathematics* 332, no. 3 (2001): 265–270.

15. M. Germano, "Differential Filters for the Large Eddy Numerical Simulation of Turbulent Flows," *Physics of Fluids* 29, no. 6 (1986): 1755–1757.

16. M. Germano, "Differential Filters of Elliptic Type," *Physics of Fluids* 29, no. 6 (1986): 1757–1758.

17. M. Girfoglio, A. Quaini, and G. Rozza, "A Finite Volume Approximation of the Navier-Stokes Equations With Nonlinear Filtering Stabilization," *Computers & Fluids* 187 (2019): 27–45.

18. M. Girfoglio, A. Quaini, and G. Rozza, "Fluid-Structure Interaction Simulations With a LES Filtering Approach in," *Communications in Applied and Industrial Mathematics* 12, no. 1 (2021): 13–28.

19. K. Kaneko and P.-H. Tsai, "NekROM", https://github.com/Nek5000/NekROM.

20. J. S. Mullen and P. F. Fischer, "Filtering Techniques for Complex Geometry Fluid Flows," *Communications in Numerical Methods in Engineering* 15, no. 1 (1999): 9–18.

21. M. A. Olshanskii and X. Xiong, “A Connection Between Filter Stabilization and Eddy Viscosity Models,” *Numerical Methods for Partial Differential Equations* 29, no. 6 (2013): 2061–2080.

22. R. Pasquetti and C. J. Xu, “Comments on Filter-Based Stabilization of Spectral Element Methods,” *Journal of Computational Physics* 182, no. 2 (2002): 646–650.

23. E. Rezaian and K. Duraisamy, “Predictive Modeling of Complex Flows using Regularized Conditionally Parameterized Graph Neural Networks,” in *American Institute of Aeronautics and Astronautics (AIAA) SCITECH 2023 Forum* (Aerospace Research Central, 2023), 1284.

24. M. Strazzullo, F. Ballarin, T. Iliescu, and C. Canuto, “New Feedback Control and Adaptive Evolve-Filter-Relax Regularization for the Navier-Stokes Equations in the Convection-Dominated Regime” arXiv preprint arXiv:2307.00675 (2023).

25. M. Strazzullo, M. Girfoglio, F. Ballarin, T. Iliescu, and G. Rozza, “Consistency of the Full and Reduced Order Models for Evolve-Filter-Relax Regularization of Convection-Dominated, Marginally-Resolved Flows,” *International Journal for Numerical Methods in Engineering* 123, no. 14 (2022): 3148–3178.

26. A. Takhirov and A. Lozovskiy, “Computationally Efficient Modular Nonlinear Filter Stabilization for High Reynolds Number Flows,” *Advances in Computational Mathematics* 44 (2018): 295–325.

27. P. H. Tsai, P. Fischer, and T. Iliescu, “A Time-Relaxation Reduced Order Model for the Turbulent Channel Flow,” *Journal of Computational Physics* 521 (2025): 113563.

28. X. Xie, F. Bao, and C. G. Webster, “Evolve Filter Stabilization Reduced-Order Model for Stochastic Burgers Equation,” *Fluids* 3, no. 4 (2018): 84.

29. L. Besabe, M. Girfoglio, A. Quaini, and G. Rozza, “Linear and Nonlinear Filtering for a Two-Layer Quasi-Geostrophic Ocean Model,” *Applied Mathematics and Computation* 488 (2025): 129121.

30. M. Girfoglio, A. Quaini, and G. Rozza, “A Novel Large Eddy Simulation Model for the Quasi-Geostrophic Equations in a Finite Volume Setting,” *Journal of Computational and Applied Mathematics* 418 (2023): 114656.

31. A. Quaini, O. San, A. Veneziani, and T. Iliescu, “Bridging Large Eddy Simulation and Reduced Order Modeling of Convection-Dominated Flows Through Spatial Filtering: Review and Perspectives arXiv preprint arXiv:2407.00231 (2024),” *Fluids* 9, no. 8 (2024): 178, https://doi.org/10.3390/fluids9080178.

32. P. H. Tsai, Parametric Model Order Reduction Development for Navier-Stokes Equations from 2d Chaotic to 3d Turbulent Flow Problems, (Ph.D. thesis, University of Illinois at Urbana-Champaign, Champaign, IL, Stati Uniti, 2023).

33. H. Xu, “Efficient Modeling of the Incompressible Flow With Moderate Large Reynolds Numbers Using a Deconvolution-Based Leray Model: Analysis, Uncertainty Quantification and Application in Aortic Dissections” 2020.

34. H. Xu, D. Baroli, F. Di Massimo, A. Quaini, and A. Veneziani, “Backflow Stabilization by Deconvolution-Based Large Eddy Simulation Modeling,” *Journal of Computational Physics* 404 (2020): 109103.

35. M. Strazzullo, F. Ballarin, T. Iliescu, and T. C. Rebollo, “Variational Multiscale Evolve and Filter Strategies for Convection-Dominated Flows arXiv preprint arXiv:2411.13957 (2024),” *Computer Methods in Applied Mechanics and Engineering* 438 (2025): 117811, https://doi.org/10.1016/j.cma.2025.117811.

36. L. Bertagna, A. Quaini, and A. Veneziani, “Deconvolution-Based Nonlinear Filtering for Incompressible Flows at Moderately Large Reynolds Numbers,” *International Journal for Numerical Methods in Fluids* 81, no. 8 (2016): 463–488.

37. F. Zoccolan, M. Strazzullo, and G. Rozza, “A Streamline Upwind Petrov-Galerkin Reduced Order Method for Advection-Dominated Partial Differential Equations Under Optimal Control,” *Computational Methods in Applied Mathematics* 25, no. 1 (2024): 237–260, https://doi.org/10.1515/cmam-2023-0171.

38. F. Zoccolan, M. Strazzullo, and G. Rozza, “Stabilized Weighted Reduced Order Methods for Parametrized Advection-Dominated Optimal Control Problems Governed by Partial Differential Equations With Random Inputs,” *Journal of Numerical Mathematics* 33, no. 1 (2024): 1–35, https://doi.org/10.1515/jnma-2023-0006.

39. D. Wells, Z. Wang, X. Xie, and T. Iliescu, “An Evolve-Then-Filter Regularized Reduced Order Model for Convection-Dominated Flows,” *International Journal for Numerical Methods in Fluids* 84, no. 10 (2017): 598–615.

40. C. Mou, E. Merzari, O. San, and T. Iliescu, “An Energy-Based Lengthscale for Reduced Order Models of Turbulent Flows,” *Nuclear Engineering and Design* 412 (2023): 112454.

41. E. Philip and E. Wong, *Sequential Quadratic Programming Methods, Technical Report NA-10-03, August* (UCSD Department of Mathematics, 2010).

42. R. B. Wilson, *A Simplicial Method for Convex Programming* (Harvard University, 1963).

43. M. Alnæs, J. Blechta, J. Hake, et al., “The Fenics Project Version 1.5,” *Archive of Numerical Software* 3, no. 100 (2015): 10–23.

44. P. Virtanen, R. Gommers, T. E. Oliphant, et al., “Scipy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods* 17, no. 3 (2020): 261–272.

45. W. Layton, C. C. Manica, M. Neda, and L. G. Rebholz, “Numerical Analysis and Computational Testing of a High Accuracy Leray-Deconvolution Model of Turbulence,” *Numerical Methods for Partial Differential Equations: An International Journal* 24, no. 2 (2008): 555–582.

46. M. Van Reeuwijk, H. J. J. Jonker, and K. Hanjalić, “Incompressibility of the Leray-α Model for Wall-Bounded Flows,” *Physics of Fluids* 18, no. 1 (2006): 1–4.

47. N. Heavner, “Locally Chosen Grad-div Stabilization Parameters for Finite Element Discretizations of Incompressible Flow Problems,” *SIAM Undergraduate Research Online (SIURO)* 7 (2017): SO1278.

48. V. DeCaria, W. Layton, A. Pakzad, Y. Rong, N. İ. Y. A. Z. İ. Sahin, and H. Zhao, “On the Determination of the Grad-div Criterion,” *Journal of Mathematical Analysis and Applications* 467, no. 2 (2018): 1032–1037.

49. V. John, A. Linke, C. Merdon, M. Neilan, and L. G. Rebholz, “On the Divergence Constraint in Mixed Finite Element Methods for Incompressible Flows,” *SIAM Review* 59, no. 3 (2017): 492–544.

50. W. Layton, C. C. Manica, M. Neda, M. Olshanskii, and L. G. Rebholz, “On the Accuracy of the Rotation Form in Simulations of the Navier–Stokes Equations,” *Journal of Computational Physics* 228, no. 9 (2009): 3433–3447.

Appendix A

Methodological Details on Opt-EFR Algorithms

This part of the Supporting Information specifies additional details on the χ-Opt-EFR and δ-Opt-EF methodologies.

The χ-Opt-EFR Algorithms: Finding an Optimal Relaxation Parameter

This section describes a strategy that combines standard optimization algorithms with the EFR strategy to find the optimal χ, starting from a fixed value of the parameter δ.

All the algorithms described in this section predict a discretized time-dependent χ, $\{\chi(t^n)\}_{n=1}^{N_T}$, minimizing an objective function. Specifically, we investigate several objective functions to deduce how the different optimization approaches would affect the accuracy of the final EFR solution. The features of the adopted strategies are summarized in Table A1.

The algorithms in Table A1 perform at each time instance Step (I), Step (II) for a given constant δ and, then, look for an optimal time-dependent $\chi(t)$ to be employed in Step (III). The process is summarized in Algorithm 2.

As mentioned in Section 3, the objective function to minimize to compute the optimal $\chi^n = \chi(t^n)$ can feature either *local* or *global* contributions. The local objective function is specialized here as follows:

$$\mathcal{L}_{\text{local}}(\chi^n) = w_{\boldsymbol{u}}\mathcal{L}^{\boldsymbol{u}}_{\text{local}}(\chi^n) + w_{\nabla\boldsymbol{u}}\mathcal{L}^{\nabla\boldsymbol{u}}_{\text{local}}(\chi^n) \tag{A1}$$

where the individual contributions are

$$\begin{aligned}\mathcal{L}^{\boldsymbol{u}}_{\text{local}}(\chi^n) &= \text{MSE}\Big(\boldsymbol{u}^n(\chi^n), \boldsymbol{u}^n_{ref}\Big)\\ \mathcal{L}^{\nabla\boldsymbol{u}}_{\text{local}}(\chi^n) &= \text{MSE}\Big(\nabla\boldsymbol{u}^n(\chi^n), \nabla\boldsymbol{u}^n_{ref}\Big)\end{aligned} \tag{A2}$$

The global objective function has the following expression:

$$\mathcal{L}_{\text{global}}(\chi^n) = w_{\boldsymbol{u}}\mathcal{L}^{\boldsymbol{u}}_{\text{global}}(\chi^n) + w_{\nabla\boldsymbol{u}}\mathcal{L}^{\nabla\boldsymbol{u}}_{\text{global}}(\chi^n) + w_p\mathcal{L}^{p}_{\text{global}}(\chi^n) \tag{A3}$$

where the global contributions are

$$\begin{aligned}\mathcal{L}^{\boldsymbol{u}}_{\text{global}}(\chi^n) &= \left|\frac{\|\boldsymbol{u}^n(\chi^n)\|^2_{L^2(\Omega)} - \|\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}}{\|\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}}\right|\\ \mathcal{L}^{\nabla\boldsymbol{u}}_{\text{global}}(\chi^n) &= \left|\frac{\|\nabla\boldsymbol{u}^n(\chi^n)\|^2_{L^2(\Omega)} - \|\nabla\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}}{\|\nabla\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}}\right|\\ \mathcal{L}^{p}_{\text{global}}(\chi^n) &= \left|\frac{\|p^{n+1}(\chi^n)\|^2_{L^2(\Omega)} - \|p^{n+1}_{ref}\|^2_{L^2(\Omega)}}{\|p^{n+1}_{ref}\|^2_{L^2(\Omega)}}\right|\end{aligned} \tag{A4}$$

Algorithm 2 presents the general pseudocode that can be applied for all the algorithms presented in Table A1, by replacing the objective function $\mathcal{L}(\chi^{n+1})$ with the specific functional of the proposed algorithm, namely (A1) or (A3).

The δ-Opt-EF Algorithms: Finding an Optimal Filter Parameter

In the present section, we provide the details on the optimized *Evolve–Filter* algorithm, namely the final velocity at time step t^n coincides with the filtered velocity: $\boldsymbol{u}^n \equiv \overline{\boldsymbol{w}}^n = \boldsymbol{u}^n(\delta^n)$. This is also equivalent to the EFR case with $\chi = 1$, and in this case, the predicted velocity would only depend on the filter parameter.

The motivation for proceeding with this analysis is that the optimization of the relaxation parameter is influenced by the choice of the filter parameter, which is kept fixed in χ-Opt-EFR. To overcome this limitation, in δ-Opt-EF we optimize the time-varying parameter $\delta(t)$ in the EF approach

TABLE A1 | Acronyms and main features of the algorithms χ-Opt-EFR. Gray cells indicate that the algorithm does not present that feature.

	Objective function type		Contributions in objective function		
Acronym	**local**	**global**	**$\boldsymbol{u}$ discrepancy**	**$\nabla\boldsymbol{u}$ discrepancy**	**p discrepancy**
χ-Opt-EFR$^{\boldsymbol{u}}_{\text{local}}$	✓	▨	✓	▨	▨
χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{local}}$	✓	▨	✓	✓	▨
χ-Opt-EFR$^{\boldsymbol{u}}_{\text{global}}$	▨	✓	✓	▨	▨
χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$	▨	✓	✓	✓	▨
χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$	▨	✓	✓	✓	✓

ALGORITHM 2 | Pseudocode for algorithms χ-Opt-EFR optimizing the relaxation parameter $\chi(t)$. Row 10 (underlined) is considered only for χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$. The optimization part is highlighted in gray.

```
1:  $\boldsymbol{u}^n_{ref}$, $n = 0, \dots, N_T - 1$; $N^u_h$; $\boldsymbol{u}^0$, $\Delta t_{opt}$, $\chi^0 = 5\Delta t$        ▷ Inputs needed
2:  for $n \in [0, \dots, N_T - 1]$ do
3:      (I) + (II) $\rightarrow \boldsymbol{w}^{n+1}$, $\overline{\boldsymbol{w}}^{n+1}$        ▷ Perform Evolve and Filter steps
4:      $\boldsymbol{u}^{n+1}_{ref} \leftarrow \boldsymbol{u}^{n+1}_{ref} \,|\, \mathbb{U}^{N^u_h}$        ▷ Project the reference field into the coarse mesh
5:      $\chi^{n+1} \leftarrow \chi^n$        ▷ Initialize$\chi$
6:      if $\text{mod}(n, \Delta t_{opt}) = 0$ then        ▷ Perform optimization every$\Delta t_{opt}$
7:          $iter = 0$
8:          repeat        ▷ SLSQP algorithm
9:              (III) $\rightarrow \boldsymbol{u}^{n+1}$        ▷ Relax step
10:             (I) $\rightarrow p^{n+2}$        ▷ (Second) Evolve step
11:             Optimize $\mathcal{L}(\chi^{n+1})$, subject to $0 \le \chi^{n+1} \le 1$
12:             $iter \leftarrow iter + 1$
13:             Output current solution: $\chi^{n+1}$
14:         until Convergence or maximum iterations reached
15:     end if
16:     Solve (III) with the optimized $\chi^{n+1} \rightarrow$ Update $\boldsymbol{u}^{n+1}$        ▷ Perform Relax step (III)
17: end for
```

directly. Moreover, we want to compare how the optimization behaves in the absence of the relaxation step, namely when only the filtered velocity is retained at each time step, so we fix $\chi = 1$.

The δ-Opt-EF strategy can be specialized in different versions, according to the objective function we decide to optimize, just as described in Section A.1 for χ-Opt-EFR. The cases taken into account are reported in Table A2. As will be seen in Section 4.1, the flow fields obtained considering the *local* objective function are inaccurate and contain spurious oscillations. For this reason, for the δ-Opt-EF objective function, we only consider the *global* optimization.

The loss type considered is the one expressed in (A3), with the contributions (A4). The difference is that $\boldsymbol{u}^{n+1} = \overline{\boldsymbol{w}}^{n+1}$ is obtained just from system (3), and, hence, the contributions will be a function of δ instead of χ, say:

$$\mathcal{L}_{\text{global}}(\delta^n) = w_{\boldsymbol{u}}\mathcal{L}^{\boldsymbol{u}}_{\text{global}}(\delta^n) + w_{\nabla\boldsymbol{u}}\mathcal{L}^{\nabla\boldsymbol{u}}_{\text{global}}(\delta^n) + w_p\mathcal{L}^{p}_{\text{global}}(\delta^n) \tag{A5}$$

with $w_{\boldsymbol{u}}$, $w_{\nabla\boldsymbol{u}}$, and w_p scalar quantities, and where the loss contributions are:

$$\begin{aligned}
\mathcal{L}^{\boldsymbol{u}}_{\text{global}}(\delta^n) &= \left| \frac{\|\boldsymbol{u}^n(\delta^n)\|^2_{L^2(\Omega)} - \|\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}}{\|\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}} \right| \\
\mathcal{L}^{\nabla\boldsymbol{u}}_{\text{global}}(\delta^n) &= \left| \frac{\|\nabla\boldsymbol{u}^n(\delta^n)\|^2_{L^2(\Omega)} - \|\nabla\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}}{\|\nabla\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}} \right| \\
\mathcal{L}^{p}_{\text{global}}(\delta^n) &= \left| \frac{\|p^{n+1}(\delta^n)\|^2_{L^2(\Omega)} - \|p^{n+1}_{ref}\|^2_{L^2(\Omega)}}{\|p^{n+1}_{ref}\|^2_{L^2(\Omega)}} \right|
\end{aligned} \tag{A6}$$

The process is outlined in Algorithm 3.

In this optimization, we consider the Kolmogorov length scale η as the initial guess at time t^0, namely, the literature common choice [36]. The optimization is performed every Δt_{opt}, as in Opt-EFR, and at each time step, the initial guess is the optimal solution at the previous time instance. The results related to this type of optimization are presented in Section 4.2.

The $\delta\chi$-Opt-EFR Algorithms: Finding the Optimal Filter and Relaxation Parameters

This final methodological section is dedicated to the *double* optimization algorithm, the $\delta\chi$-Opt-EFR.

Also in this case, we include Table A3 describing the cases investigated. Since the results obtained with the local optimizations χ-Opt-EFR$_{\text{local}}$ and δ-Opt-EF$_{\text{local}}$ are not accurate, we decide to exclude these analyses here and to consider only the global optimizations in Table A3.

The double optimization is described in Algorithm 4, where the losses have the same form as in (A3) and (A4), but with double dependency $\mathcal{L}(\delta, \chi)$. In particular, the expression of the loss can be written as:

$$\begin{aligned}
\mathcal{L}_{\text{global}}(\delta^n, \chi^n) = {} & w_{\boldsymbol{u}}\mathcal{L}^{\boldsymbol{u}}_{\text{global}}(\delta^n, \chi^n) + w_{\nabla\boldsymbol{u}}\mathcal{L}^{\nabla\boldsymbol{u}}_{\text{global}}(\delta^n, \chi^n) \\
& + w_p\mathcal{L}^{p}_{\text{global}}(\delta^n, \chi^n)
\end{aligned} \tag{A7}$$

where the global contributions are

$$\begin{aligned}
\mathcal{L}^{\boldsymbol{u}}_{\text{global}}(\delta^n, \chi^n) &= \left| \frac{\|\boldsymbol{u}^n(\delta^n, \chi^n)\|^2_{L^2(\Omega)} - \|\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}}{\|\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}} \right| \\
\mathcal{L}^{\nabla\boldsymbol{u}}_{\text{global}}(\delta^n, \chi^n) &= \left| \frac{\|\nabla\boldsymbol{u}^n(\delta^n, \chi^n)\|^2_{L^2(\Omega)} - \|\nabla\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}}{\|\nabla\boldsymbol{u}^n_{ref}\|^2_{L^2(\Omega)}} \right| \\
\mathcal{L}^{p}_{\text{global}}(\delta^n, \chi^n) &= \left| \frac{\|p^{n+1}(\delta^n, \chi^n)\|^2_{L^2(\Omega)} - \|p^{n+1}_{ref}\|^2_{L^2(\Omega)}}{\|p^{n+1}_{ref}\|^2_{L^2(\Omega)}} \right|
\end{aligned} \tag{A8}$$

In this section we will only analyze cases $w_{\nabla\boldsymbol{u}} = w_{\boldsymbol{u}} = 1$, $w_p = 0$, and $w_{\nabla\boldsymbol{u}} = w_{\boldsymbol{u}} = w_p = 1$, as those were proved to be the algorithms with the

TABLE A2 | Acronyms and main features of the algorithms δ-Opt-EF. Gray cells indicate that the algorithm does not present that feature.

	Objective function type		**Contributions in objective function**		
Acronym	**local**	**global**	**$\boldsymbol{u}$ discrepancy**	**$\nabla\boldsymbol{u}$ discrepancy**	**p discrepancy**
δ-Opt-EF$^{\boldsymbol{u}}_{\text{global}}$		✓	✓		
δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$		✓	✓	✓	
δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$		✓	✓	✓	✓

ALGORITHM 3 | Pseudocode for algorithms δ-Opt-EF optimizing the filter parameter $\delta(t)$. Row 10 (underlined) is considered only for δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$. The optimization part is highlighted in gray.

```
1:  $\boldsymbol{u}^n_{ref}$, $n = 0, \dots, N_T - 1$; $N^u_h$; $\boldsymbol{u}^0$, $\Delta t_{opt}$, $\delta^0 = \eta$        ▷ Inputs needed
2:  for $n \in [0, \dots, N_T - 1]$ do
3:      (I) $\to \boldsymbol{w}^{n+1}$                                                  ▷ Perform Evolve step
4:      $\boldsymbol{u}^{n+1}_{ref} \leftarrow \boldsymbol{u}^{n+1}_{ref} \,|\, \mathbb{U}^{N^u_h}$      ▷ Project the reference field into the coarse mesh
5:      $\delta^{n+1} \leftarrow \delta^n$                                              ▷ Initialize $\delta$
6:      if $\mathrm{mod}(n, \Delta t_{opt}) = 0$ then                                   ▷ Perform optimization every $\Delta t_{opt}$
7:          $iter = 0$
8:          repeat                                                                      ▷ SLSQP algorithm
9:              (II) $\to \boldsymbol{u}^{n+1}$                                         ▷ Filter step
10:             (I) $\to p^{n+2}$                                                       ▷ (Second) Evolve step
11:             Optimize $\mathcal{L}(\delta^{n+1})$, subject to $1 \times 10^{-5} \le \delta^{n+1} \le 1 \times 10^{-3}$
12:             $iter \leftarrow iter + 1$
13:             Output current solution: $\delta^{n+1}$
14:         until Convergence or maximum iterations reached
15:     end if
16:     Solve (II) with the optimized $\delta^{n+1} \to$ Update $\boldsymbol{u}^{n+1}$           ▷ Perform Filter step (II)
17: end for
```

TABLE A3 | Acronyms and main features of the algorithms $\delta\chi$-Opt-EFR. Gray cells indicate that the algorithm does not present that feature.

	Objective function type		Contributions in objective function		
Acronym	**local**	**global**	**u discrepancy**	**∇u discrepancy**	**p discrepancy**
$\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$	▨	✓	✓	✓	▨
$\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$	▨	✓	✓	✓	✓

ALGORITHM 4 | Pseudo-code for algorithms $\delta\chi$-Opt-EFR optimizing both parameters. Row 9 (underlined) is considered only for $\delta\chi$-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$. The optimization part is highlighted in gray.

```
1:  $u^n_{ref}$,  $n = 0, \ldots, N_T - 1$;  $N^u_h$;  $u^0$,  $\Delta t_{opt}$,  $(\delta^0, \chi^0) = (\eta, 5\Delta t)$        ▷ Inputs needed
2:  for $n \in [0, \ldots, N_T - 1]$ do
3:      (I) $\rightarrow w^{n+1}$                                                      ▷ Perform Evolve step
4:      $u^{n+1}_{ref} \leftarrow u^{n+1}_{ref} \,|\, \mathbb{U}^{N^u_h}$                 ▷ Project the reference field into the coarse mesh
        $(\chi^{n+1}, \delta^{n+1}) \leftarrow (\chi^n, \delta^n)$                        ▷ Initialize both parameters
5:      if $\mathrm{mod}(n, \Delta t_{opt}) = 0$ then                                   ▷ Perform optimization every$\Delta t_{opt}$
6:          $iter = 0$
7:          repeat
8:              (II) + (III) $\rightarrow u^{n+1}$                                      ▷ Filter and Relax step
9:              (I) $\rightarrow p^{n+2}$                                               ▷ (Second) Evolve step
10:             Optimize $\mathcal{L}(\delta^{n+1}, \chi^{n+1})$, subject to $1\times10^{-5} \le \delta^{n+1} \le 1\times10^{-3}$, $0 \le \chi^{n+1} \le 1$
11:             $iter \leftarrow iter + 1$
12:             Output current solution: $(\delta^{n+1}, \chi^{n+1})$
13:         until Convergence or maximum iterations reached
14:     end if
15:     Solve (II) + (III) with the optimized $\delta^{n+1}$ and $\chi^{n+1}$ $\rightarrow$ Update $u^{n+1}$     ▷ Perform Filter and Relax
16: end for
```

best performances in the χ-Opt-EFR and the δ-Opt-EF setting, as shown in Sections 4.1 and 4.2.

In the $\delta\chi$-Opt-EFR optimization, we consider as ranges for the parameters' search the same intervals taken into account in strategies χ-Opt-EFR and δ-Opt-EF, namely $\delta(t) \in [1\times10^{-5}, 1\times10^{-3}]$, $\chi(t) \in [0,1]$. The results related to this type of optimization are presented in Section 4.3. We remark that line 9 of Algorithm 4 is performed only if $w_p \neq 0$.

Appendix B

Additional Numerical Results

This part collects additional numerical results for the Opt-EFR strategies. Specifically, we include the following comparative studies:

- Algorithm χ-Opt-EFR (Section B.1):
  - Comparison of different χ-Opt-EFR algorithms in the L^2 relative errors with respect to the DNS reference, highlighting the improvement with respect to standard EFR;
  - A sensitivity study on how the optimization step influences the accuracy and computational time of simulations (Section B.1.1).
- Algorithm δ-Opt-EF (Section B.2):
  - An analysis on how the results change by including the incompressibility constraint into the differential filter (Section B.2.1);
  - A sensitivity study on how the optimization step influences the accuracy and computational time of simulations (Section B.2.2).

χ-Opt-EFR: Additional Results

We proceed with our analysis by investigating the accuracy with respect to the L^2 relative errors, that is, $E^u_{L^2}(t)$ and $E^p_{L^2}(t)$, displayed in Figure B1. All the results look similar in the L^2 errors, except for the χ-Opt-EFR$^{\boldsymbol{u}}_{\text{global}}$, whose solution blows up at second 1.6. Except for procedure χ-Opt-EFR$^{\boldsymbol{u}}_{\text{global}}$, all the optimized techniques yield more accurate results with respect to standard EFR (gray line in Figure B1).

However, as we can see in the graphical results of Figures 6 and 7, the L^2 relative error is not a significant metric to compare the different approaches in our test case. Indeed, even if the results for that metric are all similar, the flow dynamics are well captured in simulations of χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$, but not in the other approaches. In particular:

- In χ-Opt-EFR$^{\boldsymbol{u}}_{\text{global}}$, the field is characterized by spurious oscillations in the whole domain;
- The local methods χ-Opt-EFR$^{\boldsymbol{u}}_{\text{local}}$ and χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{local}}$ are over-diffusive.

Hence, we will not consider the L^2 errors as metrics in the remaining part of the manuscript, but we will consider the L^2 velocity and pressure norms, and the H^1 velocity seminorm to evaluate the results. We also note that the reason for analyzing *global* features, instead of *local* errors, is that, considering the optimization of local quantities, there is a phase shift in capturing the dynamics of the vortex shedding. Moreover, the local optimization is not able to fully recover all the vortices appearing in the reference DNS simulation.

On the Role of the Optimization Step

This section investigates the role of the optimization step on the efficiency and accuracy of the χ-Opt-EFR solution.

The results of the χ-Opt-EFR simulations with different k are presented in the Pareto plots in Figure B2, where the k value is visualized in the boxes next to the corresponding markers.

Figures B2 and B3 yield similar conclusions as the $\delta\chi$-Opt-EFR (Section 4.4). Specifically, all the optimized simulations show an increased accuracy with respect to standard EFR, with a similar computational time.

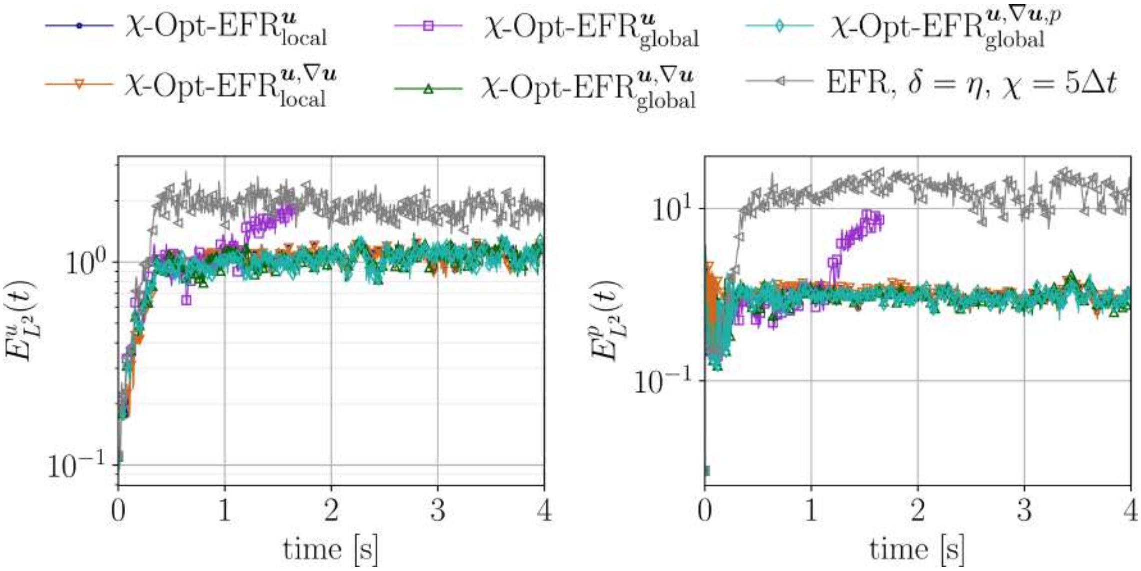


FIGURE B1 | Relative velocity and pressure errors of χ-Opt-EFR simulations with respect to the reference solutions projected on the coarse mesh. The results for the standard EFR approach are also reported.

δ-Opt-EF: Additional Results

On the Incompressibility Constraint

In EF simulations of incompressible flows, the incompressibility constraint is generally not satisfied when applying the differential filter in Step (II). In particular, the filter preserves the above-mentioned constraint under periodic boundary conditions, but not under no-slip boundary conditions [36, 45, 46]. A technique that can be exploited to tackle this problem is the so-called **grad-div stabilization**. This approach consists of penalizing the violation of the incompressibility constraint by adding to the differential filter equations a term of the form:

$$\gamma \nabla(\nabla \cdot \overline{\boldsymbol{w}^{n+1}}) \tag{B1}$$

with $\gamma = 100$, as in [47]. The reader can find more details on the grad-div stabilization in [48–50].

As stated in [25], EFR simulations with $\chi \neq 1$ are typically characterized by low values of the velocity divergence, and the grad-div stabilization is not needed. However, such stabilization becomes necessary in EF simulations, namely when $\chi = 1$.

Although the investigation in Section 4.2 does not employ the grad-div stabilization, next we highlight how the results would change in case, it is considered.

Figure B4 displays the variation in time of the space-average of the velocity divergence for the δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$, with and without grad-div stabilization. It clearly shows that in the simulations without the grad-div stabilization, the incompressibility constraint is not satisfied, while the grad-div stabilization fixes the issue.

Figures B5 and B6, similarly to Figures 11 and 12, display the time evolution of the velocity and pressure L^2 norms, and of the velocity H^1 seminorm, respectively. The plots suggest that the results with the grad-div stabilization are similar to the results obtained without stabilizing the differential filter. In particular, Figure B5 shows that the velocity norm is slightly closer to the reference norm in the last time instances of the simulation, while the pressure maintains the same chaotic behavior observed for the non-stabilized simulations.

We also provide a comparison of the global performance of simulations with and without grad-div stabilization in Table B1. The table reports the time average of the global contributions defined in Equation (A6). We can notice that the grad-div simulations provide a slightly improved velocity, as already illustrated in the left plot in Figure B5. However, the pressure accuracy is considerably lower. Moreover, if we look at the graphical representations in Figures B7 and B8, we can notice that, at the final time instance, the grad-div simulation and the DNS lift have opposite signs, while the no-grad-div simulations better capture the flow behavior.

On the Role of the Optimization Step

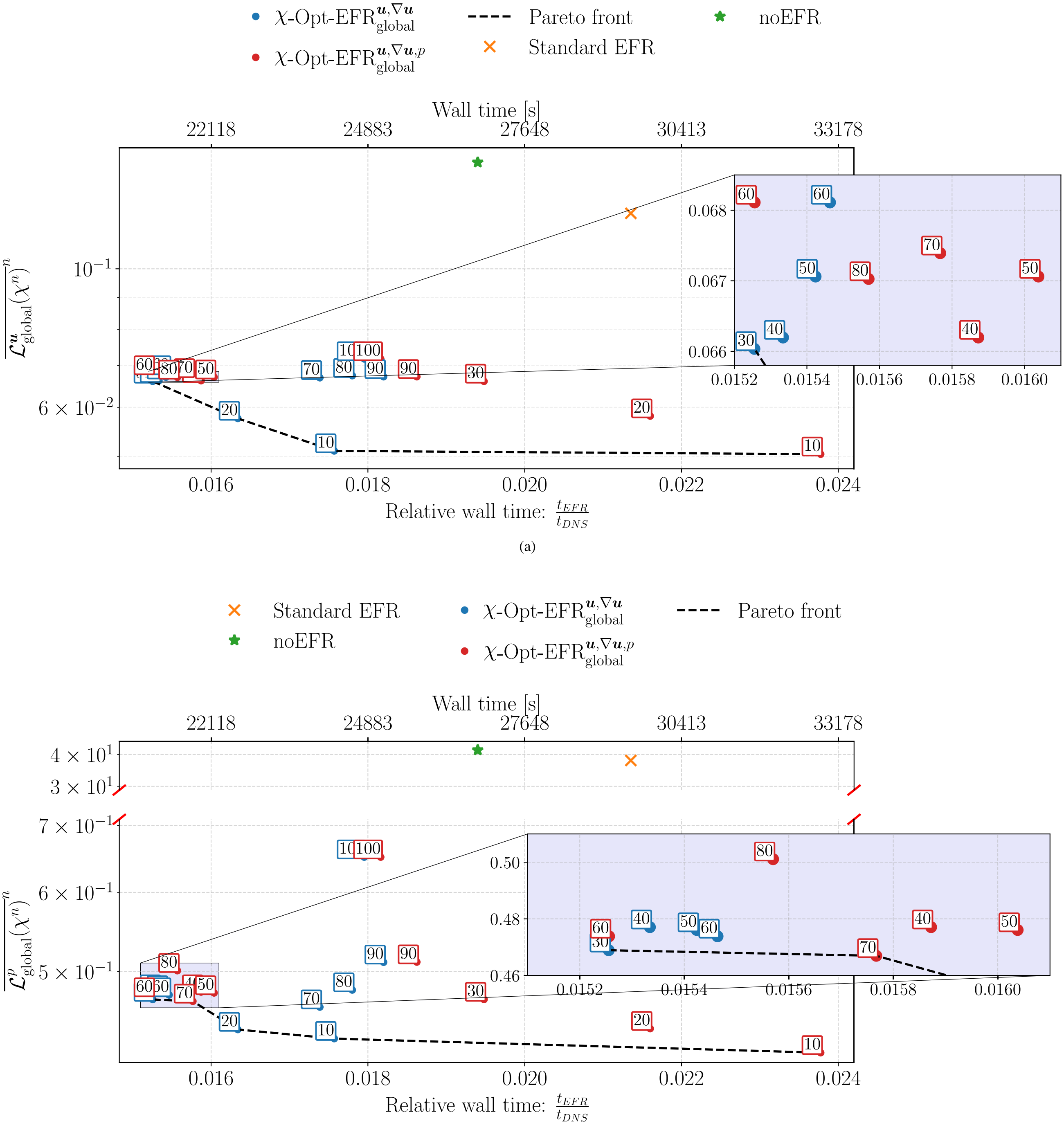
As in Section B.1.1, we include here an analysis of the accuracy depending on the optimization step.

The conclusions are the same as those outlined for the χ-Opt-EFR algorithm. Indeed, as can be seen from Figures B9 and B10, also in this case, the algorithm including the pressure contribution (δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$) is slightly more accurate, but the δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ simulations are more efficient. We can, however, notice some differences between the χ-Opt-EFR and δ-Opt-EF strategies:

i. **The accuracy of the optimized EF/EFR.** Looking at the y-scale in Figures B9 and B2, we can see that the χ-Opt-EFR algorithms provide generally more accurate results when keeping constant the filter parameter. Hence, optimizing only $\chi(t)$ and keeping the filter parameter fixed to the Kolmogorov scale produces lower errors than optimizing only $\delta(t)$ without the relaxation step.

ii. **The accuracy of the standard EF/EFR.** Standard EFR provides noisy results, while standard EF is over-diffusive but is closer to the DNS results. It is indeed more accurate than the standard EFR with $\chi \neq 1$. As an additional consideration, since it is more stable and does not exhibit spurious oscillations, the Newton method converges faster to the FE solution, and its wall clock time is considerably lower than that of the standard EFR.

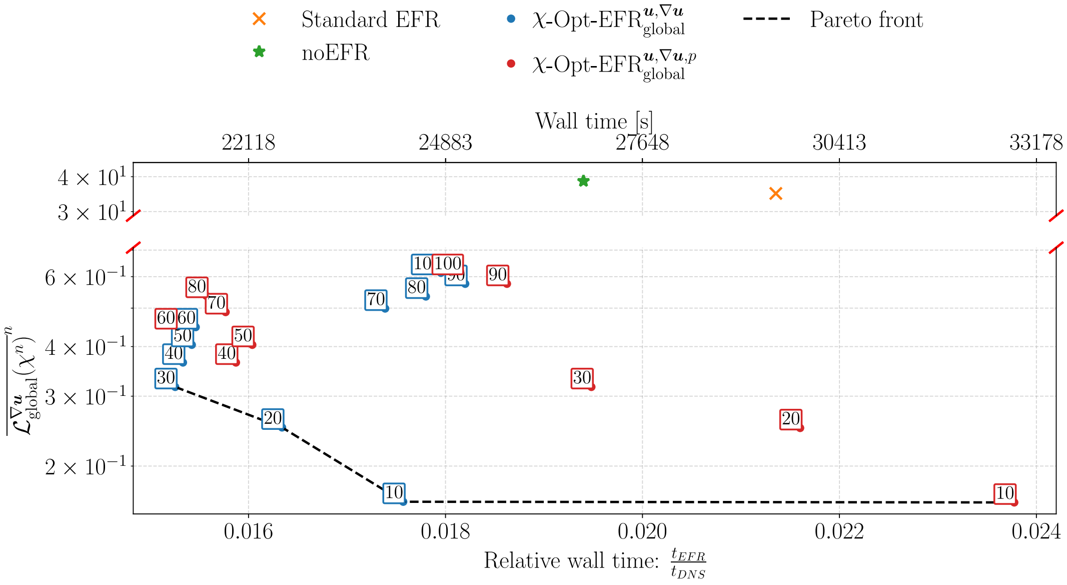
iii. **The comparison in time between standard EF and δ-Opt-EF** The δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ approach proves to be more accurate than the standard EF when considering $\Delta t_{opt} = 10\Delta t$ and $\Delta t_{opt} = 20\Delta t$. However, unlike the χ-Opt-EFR algorithms (Section 4.1), these methods are characterized by longer computational times compared to the standard EF. This is expected because EF is overdiffusive and, thus, facilitates the convergence of the Newton solver.



(a)

(b)

FIGURE B2 | Representation of how the optimization step affects the efficiency and the accuracy of the final velocity or pressure approximation in χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ simulations. The efficiency, on the x-axis, is measured through the relative wall clock time with respect to the DNS time. The accuracy, on the y-axis, is measured with respect to the L^2 velocity and pressure norms. (a) Discrepancy in the velocity L^2 norm, (b) Discrepancy in the pressure L^2 norm.



(a)

FIGURE B3 | Representation of how the optimization step affects the efficiency and the accuracy of the final velocity or pressure approximation in χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and χ-Opt-EFR$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ simulations. The efficiency, on the x-axis, is measured through the relative wall clock time with respect to the DNS time. The accuracy, on the y-axis, is measured with respect to the H^1 velocity seminorm. (a) Discrepancy on the velocity H^1 semi-norm.

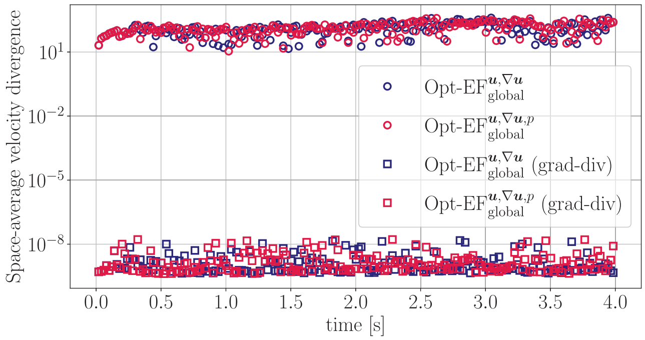


FIGURE B4 | Variation in time of the space-averaged divergence of the velocity field, for δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ with and without grad-div stabilization.

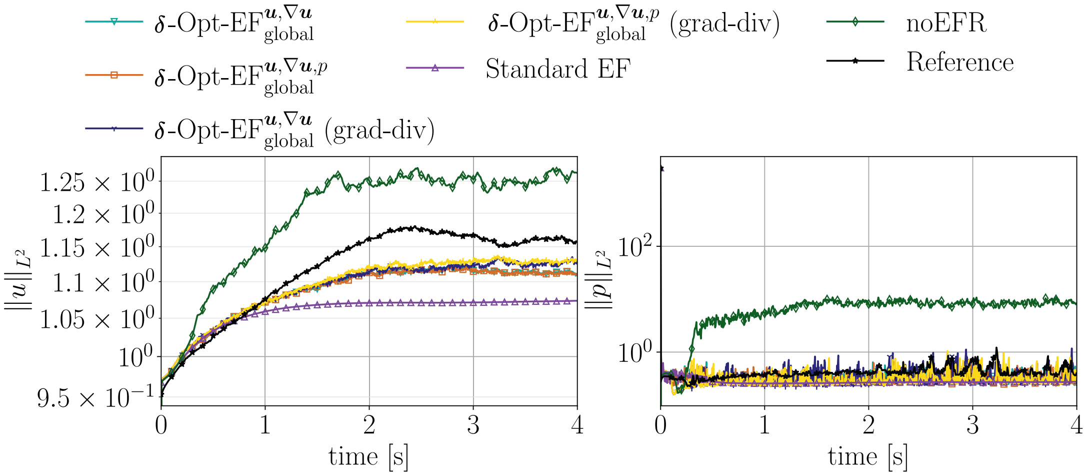


FIGURE B5 | L^2 norms of the velocity and pressure fields in δ-Opt-EF simulations, with and without grad-div stabilization.

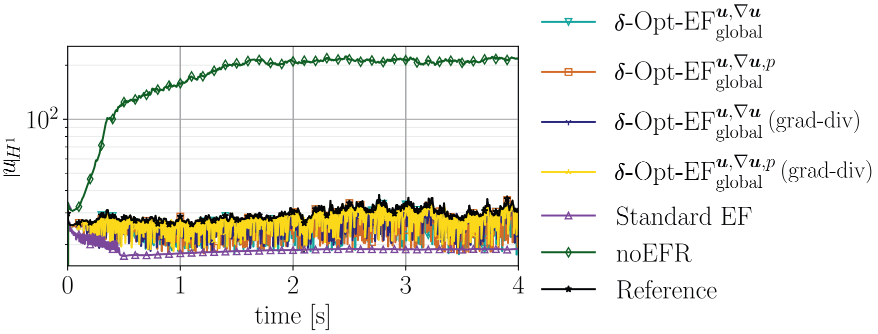


FIGURE B6 | H^1 seminorm of the velocity field in δ-Opt-EF simulations, with and without grad-div stabilization.

TABLE B1 | Comparison between simulations with and without grad-div stabilization in terms of accuracy.

		$\overline{\mathcal{L}^{u}_{\text{global}}(\delta^n)}^n$	$\overline{\mathcal{L}^{\nabla u}_{\text{global}}(\delta^n)}^n$	$\overline{\mathcal{L}^{p}_{\text{global}}(\delta^n)}^n$
Without grad-div	δ-Opt-EF$^{u,\nabla u}_{\text{global}}$	0.0578	0.188	0.420
	δ-Opt-EF$^{u,\nabla u,p}_{\text{global}}$	0.0585	0.189	0.425
With grad-div	δ-Opt-EF$^{u,\nabla u}_{\text{global}}$	0.0520	0.174	0.575
	δ-Opt-EF$^{u,\nabla u,p}_{\text{global}}$	0.0469	0.167	0.560

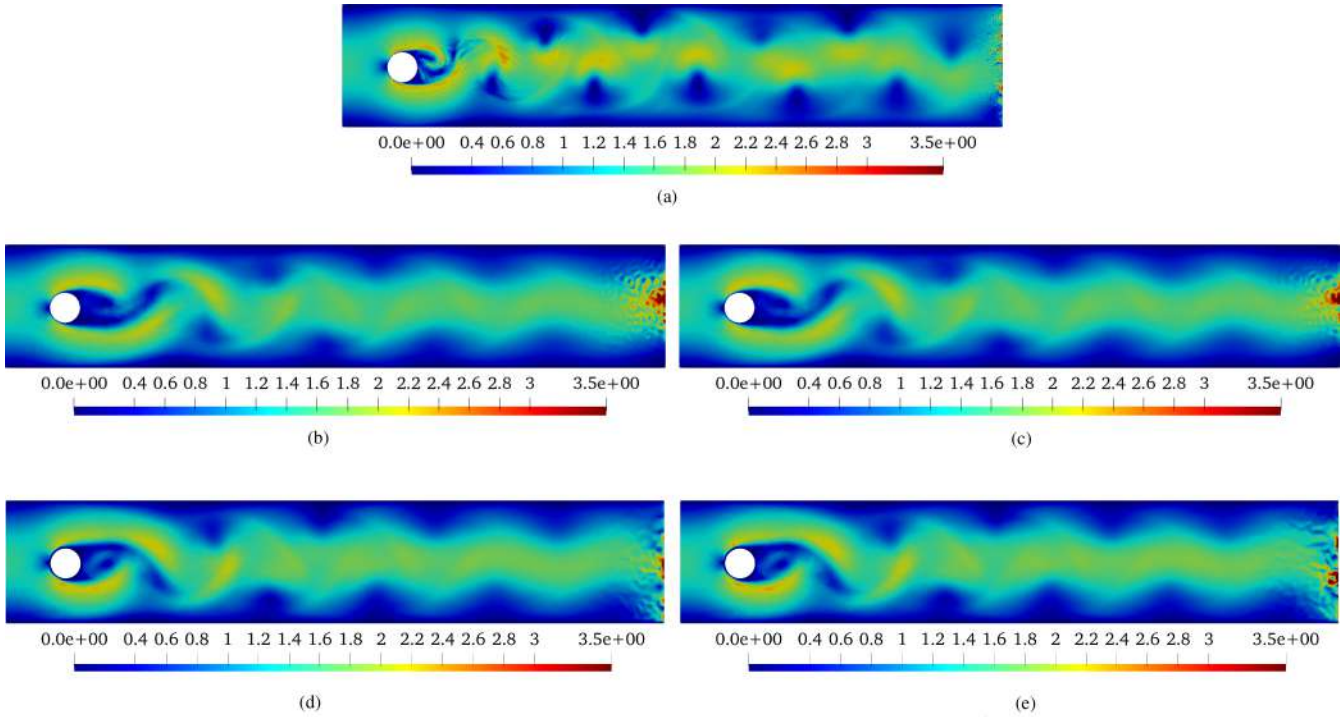


FIGURE B7 | Velocity fields for δ-Opt-EF (with and without grad-div stabilization) and reference DNS at the final time instance ($t = 4$). (a) DNS, (b) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ without grad-div, (c) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ without grad-div, (d) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ with grad-div, and (e) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ with grad-div.

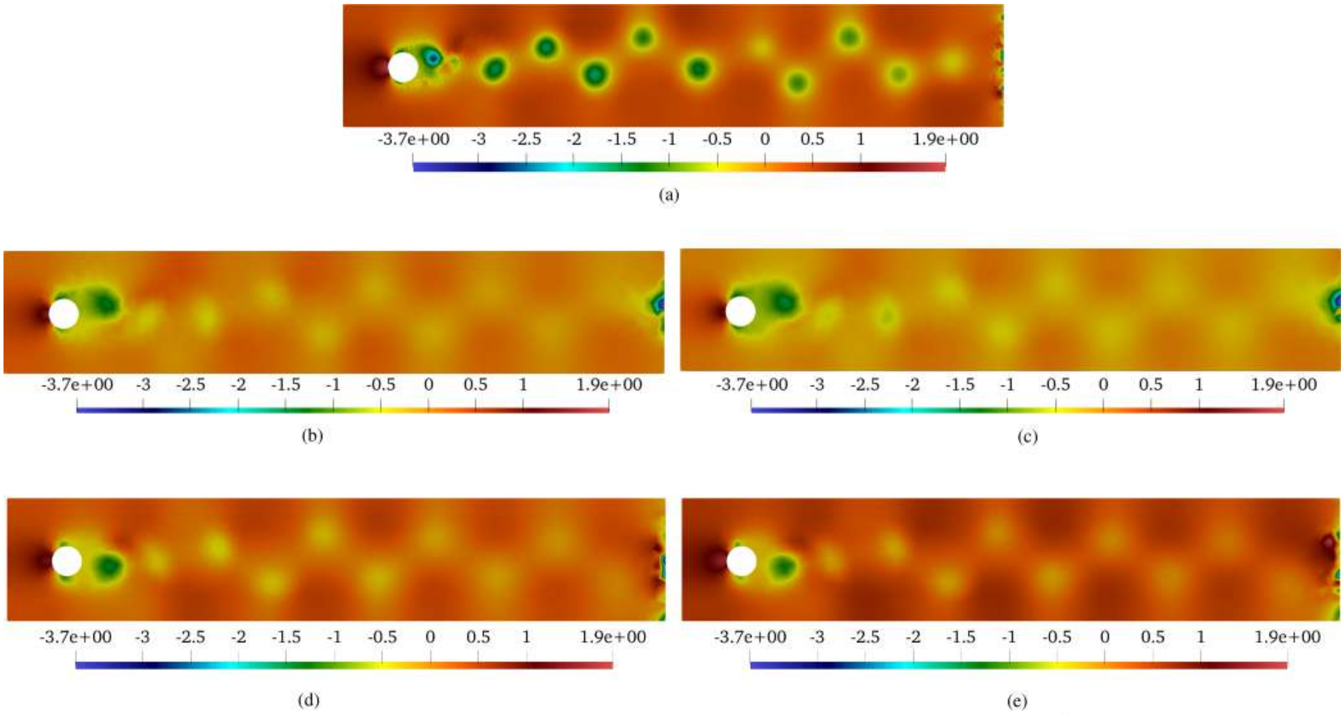
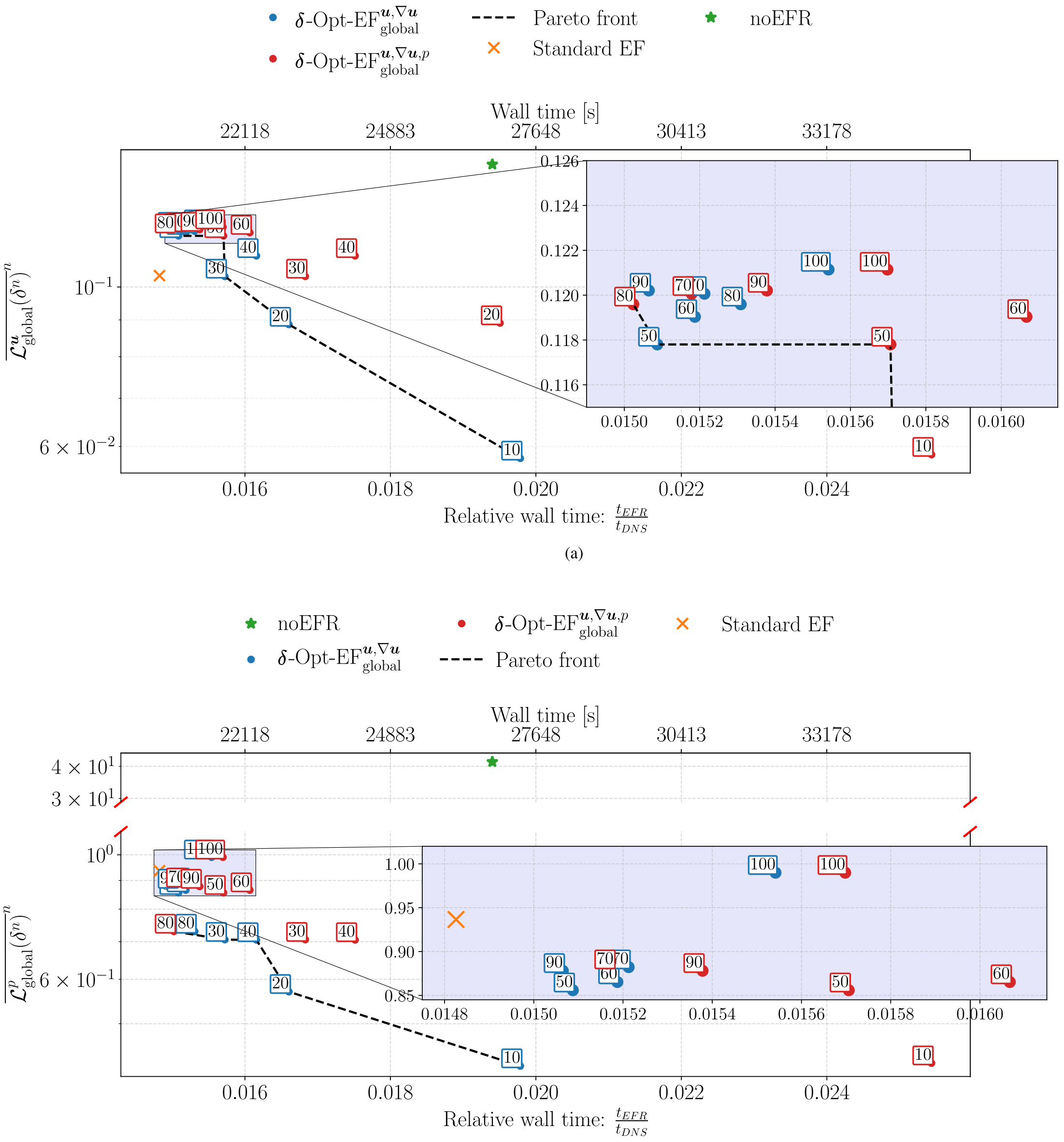


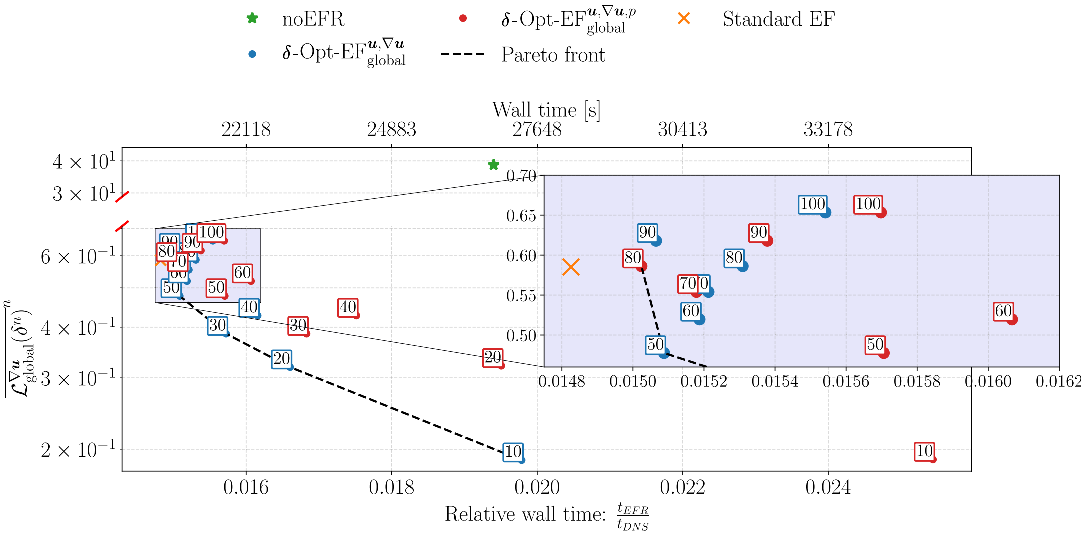
FIGURE B8 | Pressure fields for δ-Opt-EF (with and without grad-div stabilization) and reference DNS at the final time instance ($t = 4$). (a) DNS, (b) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ without grad-div, (c) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ without grad-div, (d) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ with grad-div, and (e) δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ with grad-div.



(a)

(b)

FIGURE B9 | Representation of how the optimization step affects the efficiency and the accuracy of the final velocity or pressure approximation in δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ simulations. The efficiency, on the x-axis, is measured through the relative wall clock time with respect to the DNS time. The accuracy, on the y-axis, is measured with respect to the L^2 velocity and pressure norms. (a) Discrepancy on the velocity L^2 norm, (b) Discrepancy on the pressure L^2 norm.



(a)

FIGURE B10 | Representation of how the optimization step affects the efficiency and the accuracy of the final velocity or pressure approximation in δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u}}_{\text{global}}$ and δ-Opt-EF$^{\boldsymbol{u},\nabla\boldsymbol{u},p}_{\text{global}}$ simulations. The efficiency, on the x-axis, is measured through the relative wall clock time with respect to the DNS time. The accuracy, on the y-axis, is measured with respect to the H^1 velocity seminorm. (a) Discrepancy on the velocity H^1 semi-norm.