ExaQUte

Exascale **Q**uantification of **U**ncertainties for **Te**chnology and Science Simulation

D3.3 Report of ensemble based parallelism for turbulent flows and release of solvers

Document information table

Contract number:	800898
Project acronym:	ExaQUte
Project Coordinator:	CIMNE
Document Responsible Partner:	CIMNE
Deliverable Type:	REPORT
Dissemination Level:	Confidential
Related WP & Task:	WP3, Task 3.3 & 3.4 & 3.5
Status:	Final version

Authoring

Prepared by:				
Authors	Partner	Modified	Version	Comments
Riccardo Tosi	CIMNE	All	1.0	Redaction and software development
Marc Núñez	CIMNE	All	1.0	Software development
Ramon Codina	UPC	All	1.0	Review
Jordi Pons-Prats	CIMNE	All	1.0	Review
Javier Principe	UPC	All	1.0	Redaction and review
Riccardo Rossi	CIMNE	All	1.0	Review

Change Log

Versions	Modified Page/Sections	Comments
1.0	All	Submitted version

Approval

Approved by:				
	Name	Partner	Date	OK
Task leader	Riccardo Tosi	CIMNE	30/11/2021	✓
WP leader	Javier Principe	UPC	30/11/2021	✓
Coordinator	Riccardo Rossi	CIMNE	30/11/2021	✓

Executive summary

In this work we focus on reducing the wall clock time required to compute statistical estimators of highly chaotic incompressible flows on high performance computing systems. Our approach consists of replacing a single long-term simulation by an ensemble of multiple independent realizations, which are run in parallel with different initial conditions. A failure probability convergence criteria must be satisfied by the statistical estimator of interest to assess convergence. Its error analysis leads to the identification of two error contributions: the initialization bias and the statistical error. We propose an approach to systematically detect the burn-in time in order to minimize the initialization bias, accompanied by strategies to reduce simulation cost. The framework is validated on two very high Reynolds number obstacle problems of wind engineering interest in a high performance computing environment.

Contents

Acronyms

Acronym	Meaning
BDF	backward differentiation
CAARC	Commonwealth Advisory Aeronautical Council
CPU	central processing unit
CFD	computational fluid dynamics
CFL	Courant-Friedrichs-Lewy
DNS	direct numerical simulation
FE	finite element
HPC	high performance computing
ILES	implicit large eddy simulation
Kratos	Kratos Multiphysics
LES	large eddy simulation
MC	Monte Carlo
MLMC	Multilevel Monte Carlo
NS	Navier-Stokes
OSS	orthogonjal subgrid scales
QoI	quantity of interest
Re	Reynolds number
SC	spatially correlated
SU	spatially uncorrelated
SE	statistical error
UQ	uncertainty quantification
VMS	variational multiscale

1 Introduction

The simulation of highly turbulent flows represents a well-established challenge in computational fluid dynamics (CFD), with predictions becoming more difficult as the Reynolds number (Re) increases. This situation is explained by Kolmogorov's theory which establishes that turbulent flows are characterized by multiple temporal and spatial scales, with an energy transfer cascade from larger eddies to smaller ones [50]. According to the theory, the ratio between the largest and smallest length scales is proportional to $\mathrm{Re}^{3/4}$, while the ratio between the timescales is proportional to $\mathrm{Re}^{1/2}$. This phenomenon has practical implications on flow around large objects, for example at the building scale in wind engineering. Such simulations require dealing with $\mathrm{Re} \approx 10^8$, thus implying that the smallest eddies in the flow will be around 10^6 times smaller than the largest ones with dynamics occurring at time scales around 10^5 times shorter. Such estimates effectively rule out the possibility of performing direct numerical simulations and show how even large eddy simulation (LES) approaches (either based on filtering or on numerics, see e.g. implicit large eddy simulations (ILESs) [23]) are challenging.

From a practical point of view, one has to estimate statistics of the flow, e.g. mean or variance quantities. Such estimations typically require very long simulations which include the initial transient dynamics, required for the flow to develop, followed by the effective dynamics, required for the estimator to converge. Unfortunately, despite decades of hardware improvements, such simulations require prohibitive runtimes. While the use of high performance computing (HPC) systems may reduce these runtimes, practical limits exist on the achievable speedup for a given problem size. The most important feature controlling the runtime is that time evolution in a single simulation is intrinsically sequential [40].[1]

Acknowledging such limitations, we aim at exploring an alternative strategy based on estimating statistics by averaging over numerous independent simulations, i.e. statistical ensembles. The upshot of this strategy is that each of the simulations within the ensemble can be launched independently and run in parallel, thus providing an obvious opportunity for acceleration when abundant computational resources are available. This technique has been investigated in the literature in two different settings. In one setting [37], the focus is on reducing the wall clock time on constant hardware resources. Approaches to this problem typically consist of solving linear systems with multiple right-hand sides [32, 33]. In the other setting, which our work considers, the focus is on exploiting the concurrency capabilities of HPC systems [40]. Such approaches can be seen to target the direction of next generation exascale computers, which will reach hundreds of millions of cores [1] and will follow a mega-node, kilo-core, giga-hertz rule [35].

Even though ensemble averaging has been investigated previously, the application we target (wind engineering) as well as the numerical method we employ (ILES) is significantly different from previous investigations, thus leaving the applicability of ensemble averaging unclear. The goal of our work is therefore to develop a technique to assess the efficacy of ensemble averaging when applied to any given turbulent flow problem. To this end, one of our main contributions is a statistical analysis of the approach. Complementarily, the practical question we address in this work is: How efficient is the ensemble

[1] A potential solution could be parallel-in-time methods [22], which received much attention in the last years exactly due to their potential in providing a solution for the latter problem. Unfortunately, their application does not seem to be viable in chaotic problems [56].

approach in the context of under-resolved LES methods, in particular, in wind engineering applications? Although we have targeted a specific class of engineering problems, our strategies are general and can be applied to assess the ensemble average approach for other, unrelated, problems.

In order to control bias associated to the initial conditions, the estimation of statistics of a turbulent flow entails collecting data starting only at some point in time after the flow has developed [36, 38], i.e. once the solution has been drawn to the attractor [52]. We refer to the discarded initial time interval as the *burn-in time* and the remainder as the *effective time*. When a single long simulation is performed, the burn-in time is small compared to the remaining simulation time, which contains the effective dynamics. Unfortunately, this is not the case when the same amount of simulation time is distributed across an ensemble. Indeed, the same burn-in time will be paid by all realizations in the ensemble and, as a consequence, the total effective time will be reduced. The reduction of the burn-in time is therefore key to making ensemble averaging feasible. Our statistical model provides the tools to analyse the bias associated with the initial conditions, thus allowing us to faithfully select a practical burn-in time.

One of the additional research questions we address is what to assign as a distribution for the initial conditions. The aim here is again to control the burn-in time. To this end, two different types of Gaussian random vector fields are considered; white noise perturbations and spatially-correlated solenoidal fields. Another approach to cost reduction which we explore is to use a less accurate and less expensive time integration procedure during the burn-in phase, e.g. by increasing the time step size.

The final question we seek to address is how long each realization should be. Increasing the number of realizations improves concurrency but also increases the aggregated burn-in time, so there is a trade-off that needs to be considered. The present study evaluates this trade-off in the case of flows around bodies. In this work, we demonstrate that very short simulations are sufficient, thus making the approach highly efficient for the class of problems we have targeted.

Our approach is validated with two numerical examples. The first example consists of wind flowing past a rectangular obstacle. The second example consists of wind flowing around a high-rise building. To conduct this study, we used the uncertainty quantification library XMC [2], the finite element software library Kratos Multiphysics (Kratos) [16, 17, 42], and the distributed environment framework PyCOMPSs [3, 39, 53].

The remainder of the article is structured as follows. In section 2 we describe the ILES method we use and, in section 3, we describe our statistical approach. A set of numerical experiments evaluating the performance of ensemble averaging is presented in section 4. A software release containing the methods presented in this report is presented in section 5. Concluding remarks close the work in section 6.

2 Implicit Large eddy simulation

In this section we briefly describe the ILES model we use for the simulation of turbulent flows around obstacles. We restrict ourselves to incompressible flows which cover a wide range of applications in wind engineering. In strong form, the Navier-Stokes (NS) problem for an incompressible fluid consists of finding a velocity $\boldsymbol{u}$ and a pressure p defined in a bounded domain $D \subseteq \mathbb{R}^d$, where $d = 2, 3$ is the number of space dimensions, as the solution

of

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } [0,T] \times D, \tag{1a}$$
$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } [0,T] \times D, \tag{1b}$$

where $\boldsymbol{f}$ is the force vector and ν the kinematic viscosity (as usual, we denote vectors and tensors using bold characters). These equations must be complemented with appropriate boundary conditions, which can be of Dirichlet type, i.e. $\boldsymbol{u} = \boldsymbol{u}_g$ applied on Γ_{D}, or of Neumann type, i.e. $(-p\mathbf{I} + \nu \nabla \boldsymbol{u}) \cdot \boldsymbol{n} = \boldsymbol{t}_N$ applied on Γ_{N}, for all $t \in [0,T]$. In this section we consider $\partial D = \Gamma_{\mathrm{D}}$ and $\boldsymbol{u}_g = 0$ to simplify the notation but in general $\Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N}} = \partial D$ and $\Gamma_{\mathrm{D}} \cap \Gamma_{\mathrm{N}} = \emptyset$, which is the case in the numerical experiments of section 4. Equations (1a) and (1b) must be supplemented with appropriate initial conditions.

As usual, we denote by $L^p(D)$, $1 \leq p < \infty$, the spaces of functions whose p-th power is Lebesgue integrable in D. The space of functions whose first-order distributional derivatives are in $L^2(D)$ and have zero trace on ∂D is denoted by $H_0^1(D)$ and its topological dual by $H^{-1}(D)$. We write $(\cdot,\cdot)$ to denote the integral over D of the product of any two functions f and g, whenever it makes sense. Given a Banach space X, $L^p(0,T;X)$ denotes the space of functions whose X-norm is in $L^p(0,T)$ whereas $\mathcal{D}'(0,T;X)$ denotes the space of distributions in time with values in X.

The weak form of the NS problem consists of finding $\boldsymbol{u} \in L^2(0,T;V)$ and $p \in \mathcal{D}'(0,T;Q)$ such that

$$(\partial_t \boldsymbol{u}, \boldsymbol{v}) + (\boldsymbol{u} \cdot \nabla \boldsymbol{u}, \boldsymbol{v}) + \nu(\nabla \boldsymbol{u}, \nabla \boldsymbol{v}) - (p, \nabla \cdot \boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}), \quad \text{for all } \boldsymbol{v} \in V, \tag{2a}$$
$$(q, \nabla \cdot \boldsymbol{u}) = 0, \quad \text{for all } q \in Q, \tag{2b}$$

where $V = H_0^1(D)^d$ and $Q = L_0^2(D) := L^2(D)/\mathbb{R}$ (L^2 functions with zero mean).

The discrete form of equations (2a) and (2b) is obtained in the framework of the variational multiscale (VMS) method, which incorporates LES concepts in the numerics, giving an ILES method for the simulation of turbulent incompressible flows. The VMS method was originally introduced in [26, 27] as a framework for the development of stabilization methods, which are designed to overcome the two main problems of the numerical approximation of equations (2a) and (2b). The first one is the compatibility required between the velocity and pressure spaces which need to satisfy an *inf-sup* condition to guarantee the stability of the approximation. The second one is the lack of robustness of the Galerkin method in the advection dominated regime. There are many VMS methods and a complete review of them is out of the scope of this article, see e.g. [12]; we briefly describe our choice in what follows.

The starting point of VMS formulations is a splitting of the solution space as $V = V_h \oplus \widetilde{V}$, into a finite element (FE) space V_h and a space of subgrid scales $\widetilde{V}$. The FE space V_h is built on top of a partition $\mathcal{T}_h$ of the domain D and is used to represent resolvable scales. In this way, a function $\boldsymbol{u} \in V$ is decomposed as $\boldsymbol{u} = \boldsymbol{u}_h + \widetilde{\boldsymbol{u}}$. The same splitting can be considered for the pressure space although this is not necessary to develop stable methods and the simplest approach is to consider $\widetilde{p} = 0$. Alternatively, a model for the pressure subscale depending on the velocity divergence is commonly used, see e.g. [15, 31].

The VMS decomposition of the test function $\boldsymbol{v}$ in equations (2a) and (2b) gives rise to an equation for the resolved scales (tested by $\boldsymbol{v}_h$) and an equation for the fine scales (tested by $\widetilde{\boldsymbol{v}}$). However, these two equations are coupled and, because the space $\widetilde{V}$ is infinite-dimensional, some modelling assumptions for the fine scale equation are required

to close the system. This modeling step is the algebraic approximation of the differential operator acting on the fine scales,

$$((\boldsymbol{u}_h + \widetilde{\boldsymbol{u}}) \cdot \nabla \widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{v}}) - \nu(\nabla \widetilde{\boldsymbol{u}}, \nabla \widetilde{\boldsymbol{v}}) \approx (\tau^{-1} \widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{v}}), \tag{3}$$

where τ is a piecewise constant function, computed within each element $K \in \mathcal{T}_h$ as

$$\tau_K^{-1} = \frac{c_1 \nu}{h_K^2} + \frac{c_2 \|\boldsymbol{u}_h + \widetilde{\boldsymbol{u}}\|_K}{h_K}. \tag{4}$$

Here, h_K is a characteristic length of K, c_1 and c_2 are algorithmic constants that depend only on the degree of the finite element approximation being used, and $\|\cdot\|_K$ is some norm defined on each element, e.g. the $L^2(K)$-norm. Equation (4) can be motivated also by a heuristic Fourier analysis argument [11], although the important point is its asymptotic behavior in terms of h_K, ν and $\|\boldsymbol{u}_h + \widetilde{\boldsymbol{u}}\|_K$. Some further modelling choices lead to different VMS models. These choices include:

- *Static/Dynamic subscales:* from the VMS decomposition it follows that $\partial_t \boldsymbol{u} = \partial_t \boldsymbol{u}_h + \partial_t \widetilde{\boldsymbol{u}}$. Considering *dynamic subscales*, introduced in [11, 13], has some advantages like a correct behavior of time integration schemes and better accuracy. In particular, stability and convergence for the Stokes problem can be proved without any restriction on the time step size and the stabilization parameters on which the formulation depends. The typical approach, however, is the use of *quasistatic subscales* to neglect $\partial_t \widetilde{\boldsymbol{u}}$.

- *Linear/Nonlinear subscales:* applying the VMS decomposition to the nonlinear convective term, four different contributions are obtained on each equation (fine and coarse), that is, $\boldsymbol{u} \cdot \nabla \boldsymbol{u} = (\boldsymbol{u}_h + \widetilde{\boldsymbol{u}}) \cdot \nabla (\boldsymbol{u}_h + \widetilde{\boldsymbol{u}})$. After the approximation in equation (3) it is possible to keep all the contributions, as proposed in [11, 13]. A simpler alternative is to perform the approximation $\boldsymbol{u} \cdot \nabla \boldsymbol{u} \approx \boldsymbol{u}_h \cdot \nabla \boldsymbol{u} + \widetilde{\boldsymbol{u}}$ (thus neglecting $\widetilde{\boldsymbol{u}}$ in equation (4) and the quadratic term in $\widetilde{\boldsymbol{u}}$ in equation (3)) which is enough to have numerical stability.

- *The space of subscales:* the choice of a space for the approximation of the subscales defines a projector $\mathcal{P}$ to be used in the fine scale equation. One option is to choose $\widetilde{V}$ as the space of the residual, that is to simply take $\mathcal{P} = \mathbb{I}$ (the identity). We refer to this space of subscales as the *algebraic subscales*. Another possibility is to consider the space of the subscales orthogonal to the FE space, that is, to take $\mathcal{P} := \Pi_h^{\perp} = \mathbf{I} - \Pi_h$, where Π_h is the projection onto the FE space [11].

A complete assessment of these modelling choices can be found in [14]. In this work we use static, linear, orthogonal subscales. Using nonlinear and/or dynamic subscales requires tracking them along the iterative and time integration loops, with the consequent increase in memory demands and computational cost (the simplest option is to store the subscales at the integration points). Although using dynamic, nonlinear orthogonal subscales provides a better accuracy, these subscales also imply a higher computational cost. The evaluation of this problem-dependent trade-off is outside the scope of this article.

However, even if it is simpler to consider algebraic subscales, orthogonjal subgrid scales (OSS) enjoy a number of important properties that are worth having, such as

stability without restrictions on the time step size [13], a clear scale separation in the energy transfers and the possibility of predicting backscatter with a stable numerical method [51] and convergence towards weak solutions [4]. It is worth mentioning that, after the introduction of OSS a number of projection-based method appeared in which only some terms involving fine scales are kept in the resolved scale equation, see [12, section 4.1]. The method we use in this article belongs to this class. It retains only the advective and pressure terms and projects them separately, so it is usually referred to as term-by-term OSS.

With all these modelling choices the final semi-discrete problem to be solved consists of finding $\boldsymbol{u}_h \in V_h$ and $p_h \in Q_h$ such that

$$\begin{aligned}(\partial_t \boldsymbol{u}_h, \boldsymbol{v}_h) + (\boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h, \boldsymbol{v}_h) + \nu(\nabla \boldsymbol{u}_h, \nabla \boldsymbol{v}_h) - (p_h, \nabla \cdot \boldsymbol{v}_h) + (q_h, \nabla \cdot \boldsymbol{u}_h) \\ + (\boldsymbol{u}_h \cdot \nabla \boldsymbol{v}_h, \tau \mathcal{P}(\boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h)) + (\nabla q_h, \tau \mathcal{P}(\nabla p_h)) = (\boldsymbol{f}, \boldsymbol{v}_h),\end{aligned} \tag{5}$$

for any $\boldsymbol{v}_h \in V_h$ and $q_h \in Q_h$.

Equation (5) is integrated in time using a second-order backward differentiation (BDF) scheme. The projections onto the FE space are handled explicitly, that is, given $\boldsymbol{u}_h^n$ we first obtain the projection of the convective term η_h (which satisfies $(q_h, \eta_h) = (q_h, \boldsymbol{u}_h^n \cdot \nabla \boldsymbol{u}_h^n)$ for any $q_h \in L^2$) and the projection of the pressure ξ_h (which satisfies $(q_h, \xi_h) = (\nabla q_h, \nabla p_h^n)$ for any $q_h \in L^2$). We use linear finite elements for the velocity, the pressure and the projections.

The final system is solved iteratively by a predictor-corrector scheme with a block preconditioner which permits us to separate the computation of the velocity and the pressure variable. It is a variant of the classical fractional step method [9, 10] obtained algebraically [20]. This algebraic view opens the door to other options, like performing an iterative correction, eventually converging to the monolithic solution.

3 Statistical analysis

In this section, we introduce our statistical framework. In section 3.1 we define the statistical operators and the probability criteria. Different sources of error are identified and analyzed in section 3.2. We discuss the generation of initial velocity fields, which provides independent ensemble realizations in section 3.3. Finally, considerations on how to determine the length of the burn-in time are provided in section 3.4.

3.1 Problem outline

Let $u(t, \boldsymbol{x}, w)$ denote the solution to equations (1a) and (1b), where the initial condition $u_0(\boldsymbol{x}) = u_0(\boldsymbol{x}; w)$, $w \in \Omega$, is a random field over D. In this setting, $u(t, \boldsymbol{x}, w)$ is a random field over $(0, T) \times D$. From u, we can compute the quantity of interest $\mathrm{Q}(t, w) := \mathrm{Q}(u(t, \boldsymbol{x}, w))$, which is a stochastic process. For ease of notation, we omit the dependency on w in what follows, except when it is needed to clarify the presentation. The expected value of a process X is denoted $\mathbb{E}[X]$, its variance is denoted $\mathbb{V}[X]$ and its covariance with a process Y is denoted $\mathrm{co}\mathbb{V}[X, Y]$. These quantities can be estimated by sample averaging. For instance, for any N independent samples, $X_1, \ldots, X_N$, we define the sample mean $\mathrm{E}^N[X] = \frac{1}{N} \sum_{n=1}^N X_n$ and the sample variance $\mathrm{V}^N[X] = \frac{1}{N-1} \sum_{n=1}^N (X_n - \mathrm{E}^N[X])^2$.

Our goal is to compute the long-term expected value of $\mathbb{E}[\mathrm{Q}]$, that is,

$$\overline{\mathrm{Q}} = \lim_{t\to\infty} \mathbb{E}[\mathrm{Q}(t)]. \tag{6}$$

Assuming ergodicity, $\overline{\mathrm{Q}}$ can also be computed as the time average, defined as

$$\langle \mathrm{Q} \rangle = \lim_{T\to\infty} \frac{1}{T} \int_0^T Q(t)\,\mathrm{d}t. \tag{7}$$

Time averages performed over the effective time $[T_0, T]$ are defined as

$$\langle \mathrm{Q} \rangle_{T_0,T} = \frac{1}{T - T_0} \int_{T_0}^T Q(t)\,\mathrm{d}t, \tag{8}$$

and the notation in equation (8) simplifies to $\langle \mathrm{Q} \rangle_T$ when $T_0 = 0$, that is $\langle \mathrm{Q} \rangle_T := \langle \mathrm{Q} \rangle_{0,T}$.

Given N realizations of the process Q, $\mathrm{Q}_n := \mathrm{Q}(t, w^{(n)})$, for $n = 1, ..., N$, we aim to approximate $\overline{\mathrm{Q}}$ by

$$\overline{\mathrm{Q}} \approx \mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}] = \frac{1}{N} \sum_{n=1}^{N} \langle \mathrm{Q}_n \rangle_{T_0,T}. \tag{9}$$

Using this notation, the statistical problem we face is to find the optimal values of N, T_0, T (those that minimize the computational cost and/or the time to solution) while satisfying the probability convergence criteria

$$\mathbb{P}\left[\left|\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}] - \overline{\mathrm{Q}}\right| > \varepsilon\right] \leq \phi,\ \varepsilon > 0,\ \phi \ll 1, \tag{10}$$

where ε is the desired tolerance and $1 - \phi$ the confidence on the sampled statistical estimator. Such a condition requires that the probability of the error exceeding ε should not be greater than ϕ.

3.2 Error analysis

There are two sources of error in equation (9). First, the choice of the random initial condition may not be compatible with the long-term statistically stationary, thus triggering a transient perturbation with a nonvanishing contribution to the mean, sometimes called initialization bias [5]. The use of a finite number of samples of finite duration is a second source of error. Increasing the number of samples, as well as the length of their effective time intervals, will also reduce the influence of the initialization bias.

3.2.1 Initialization bias

To analyze the error in approximation (9) we decompose each realization Q_n into two components,

$$\mathrm{Q}_n = A_n + S_n, \tag{11}$$

where S_n is the realization of an ergodic and stationary process S, and A_n is a transient perturbation. This means to assume $\lim_{t\to\infty} \mathbb{E}[A_n(t)] = 0$ or, equivalently,

$$\lim_{t\to\infty} \mathbb{E}[\mathrm{Q}_n(t)] = \overline{\mathrm{Q}} = \mathbb{E}[S]. \tag{12}$$

In this setting,

$$\mathbb{E}[\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]] = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}[\langle \mathrm{Q}_n \rangle_{T_0,T}] = \frac{1}{N}\sum_{n=1}^{N} \left(\mathbb{E}[\langle S_n \rangle_{T_0,T}] + \mathbb{E}[\langle A_n \rangle_{T_0,T}]\right). \tag{13}$$

Since S is statistically stationary,

$$\mathbb{E}[\langle S_n \rangle_{T_0,T}] = \mathbb{E}\left[\frac{\int_{T_0}^{T} S_n(t)\,\mathrm{d}t}{T-T_0}\right] = \frac{1}{T-T_0}\int_{T_0}^{T} \mathbb{E}[S_n]\,\mathrm{d}t = \mathbb{E}[S_n] = \mathbb{E}[S] = \overline{\mathrm{Q}}. \tag{14}$$

Therefore,

$$\mathbb{E}[\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]] = \overline{\mathrm{Q}} + \mathrm{B}, \tag{15}$$

where

$$\mathrm{B} = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}[\langle A_n \rangle_{T_0,T}] \tag{16}$$

is the bias of the initial condition, which can be mitigated if $\mathbb{E}[\langle A_n \rangle_{T_0,T}]$ decays sufficiently fast. If

$$\int_0^{\infty} |\mathbb{E}[A_n](t)|\ \mathrm{d}t < \infty, \tag{17}$$

then we have that

$$|\mathbb{E}[\langle A_n \rangle_{T_0,T}]| = \frac{1}{T-T_0}\left|\int_{T_0}^{T} \mathbb{E}[A_n](t)\,\mathrm{d}t\right| \leq \frac{1}{T-T_0}\int_{T_0}^{T} |\mathbb{E}[A_n](t)|\ \mathrm{d}t \tag{18}$$

$$\leq \frac{1}{T-T_0}\int_0^{\infty} |\mathbb{E}[A_n](t)|\ \mathrm{d}t \xrightarrow[T-T_0\to\infty]{} 0. \tag{19}$$

In other words, if equation (17) holds, then increasing $T - T_0$ eventually decreases $|\mathrm{B}|$.

An estimation of the decay rate can be made under stronger assumptions on the transient perturbation. For illustration purposes we consider a fast decay of the form

$$A_n^f(t) = A_{0,n}^f \mathrm{e}^{-\frac{t}{\tau}}, \tag{20}$$

which is an example of exponentially ergodic processes. For a wide class of stochastic processes satisfying a dissipativity condition, it can be proved that the transient perturbation decays exponentially [45, theorem 6.1], i.e. it satisfies

$$|\,\mathbb{E}[A^f]| = |\,\mathbb{E}[A_0^f]|\mathrm{e}^{-\frac{t}{\tau}}, \tag{21}$$

as in [21, equation (3)].

If we now consider its time average we get

$$\mathbb{E}[\langle A_n^f \rangle_{T_0,T}] = \mathbb{E}[A_{0,n}^f]\langle \mathrm{e}^{-\frac{t}{\tau}} \rangle_{T_0,T} = \tau\,\mathbb{E}[A_{0,n}^f]\left(\frac{\mathrm{e}^{-T_0/\tau} - \mathrm{e}^{-T/\tau}}{T-T_0}\right), \tag{22}$$

from where we see a decay of the form

$$\mathbb{E}[\langle A_n^f \rangle_{T_0,T}] = \mathcal{O}((T-T_0)^{-1}). \tag{23}$$

Therefore, $\mathbb{E}[\langle A_n^f \rangle_{T_0,T}]$ is a decreasing function of $T - T_0$. However, as it can be seen in equation (22), it is also decreasing when $T - T_0$ is kept constant while T and T_0 separately increase. In practice, T is fixed so increasing T_0 decreases $T - T_0$; this is the trade-off we analyse in the examples of section 4.

3.2.2 Statistical error

The previous analysis makes it clear that bias can be reduced by increasing $T - T_0$ and, for specific transient perturbations, increasing T_0. However, to assess statistical accuracy, equation (10) needs to be evaluated. Given a bound $\left|\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}] - \mathbb{E}[\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}]]\right| \leq \varepsilon$, the asymptotic normality of the estimator $\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}]$, in the limit $N \to \infty$, implies that [19, chapter 3]

$$\left|\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}] - \mathbb{E}[\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}]]\right| \leq \mathcal{C}_\phi \sqrt{\mathbb{V}[\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}]]} \leq \varepsilon, \tag{24}$$

with probability $1 - \phi$ as the tolerance $\varepsilon \to 0$. $\mathcal{C}_\phi$ is the confidence coefficient defined as $\mathcal{C}_\phi = \Phi^{-1}(1 - \frac{\phi}{2})$, there Φ is the cumulative distribution function of a standard normal distribution. The total error in equation (10) can then be bounded with confidence $1-\phi$, as follows,

$$\begin{aligned}\left|\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}] - \overline{\mathrm{Q}}\right| &\leq \left|\overline{\mathrm{Q}} - \mathbb{E}[\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}]]\right| + \left|\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}] - \mathbb{E}[\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}]]\right| \\ &\leq \left|\overline{\mathrm{Q}} - \mathbb{E}[\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}]]\right| + \mathcal{C}_\phi \sqrt{\mathbb{V}[\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}]]}.\end{aligned} \tag{25}$$

We define the statistical error (SE) to be $\mathrm{SE} = \sqrt{\mathbb{V}[\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}]]}$. Thus, using equation (15), we get

$$\left|\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}] - \overline{\mathrm{Q}}\right| \leq |\mathrm{B}| + \mathcal{C}_\phi \mathrm{SE}, \tag{26}$$

where B is the initialization bias defined in equation (16). For a given confidence $1 - \phi$, the probability convergence criteria then reads

$$|\mathrm{B}| + \mathcal{C}_\phi \mathrm{SE} \leq \varepsilon. \tag{27}$$

The bias error $|\mathrm{B}|$ was analyzed above, let us now focus on the SE term. Assuming each A_n and S_n are independent, we have

$$\mathbb{V}[\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}]] = \frac{1}{N^2}\sum_{n=1}^{N} \mathbb{V}[\langle S_n\rangle_{T_0,T}] + \frac{1}{N^2}\sum_{n=1}^{N} \mathbb{V}[\langle A_n\rangle_{T_0,T}] + \frac{1}{N^2}\sum_{\substack{n,m=1 \\ n\neq m}}^{N} \mathrm{co}\mathbb{V}[\langle \mathrm{Q}_n\rangle_{T_0,T}, \langle \mathrm{Q}_m\rangle_{T_0,T}]. \tag{28}$$

The first term on the right-hand side of equation (28) can be written as

$$\mathbb{V}[\langle S_n\rangle_{T_0,T}] = 2\frac{\mathbb{V}[S]}{T - T_0}\int_{T_0}^{T} \left(1 - \frac{t}{T - T_0}\right) \rho(t)\, \mathrm{d}t, \tag{29}$$

where ρ is the autocorrelation function, see [50, problem 3.37]. The long-time limit of the integral in equation (29) is the integral time scale of the process [50, section 3.6], which is a correlation constant associated to the quantity of interest. Therefore, $\mathbb{V}[\langle S_n\rangle_{T_0,T}]$ decays like $(T - T_0)^{-1}$.

The second term on the right-hand side of equation (28) cannot be estimated without making assumptions on the behavior of the transient perturbation A. If we consider the same fast decay of the previous subsection, equation (20), a straightforward computation shows that $\mathbb{V}[\langle A_n^f\rangle_{T_0,T}] = \mathcal{O}((T - T_0)^{-2})$.

The last term in equation (28) depends on the correlation between realizations. An example of the effects of the correlation between realizations of turbulent flow in a channel

is presented in [40]. If these realizations are independent, the final term in equation (28) is negligible and the dominant term in equation (27) depends on the decay rate of the transient perturbations A_n. To this end, we discuss two initial condition strategies which help to provide independent realizations in section 3.3.

If the decay of the transient perturbation is slower than equation (20), the decay of the statistical error will be dominated by the second term in equation (28). Therefore, both the bias and statistical error will decay at the same rate, and the left-hand side of equation (27) will decay like $(T - T_0)^{-q}$ for some $q < 1$. If the decay of the transient perturbation is fast (equation (20) holds), the overall error will be dominated by the first term in equation (28), and the left-hand side of equation (27) will decay like $N^{-0.5}$ and $(T - T_0)^{-0.5}$. In the numerical experiments presented in section 4, we verify the decay rates to assess if the initialization bias can be neglected in computing the error.

The left hand side of equation (26) is estimated by approximating the variance by the sample variance. Because we aim at computing the variance of an average, we perform K repetitions of each experiment, totalling $K \cdot N$ independent simulations. In this way, the right hand side of equation (26) can be approximated as follows,

$$\mathcal{C}_\phi \sqrt{\mathbb{V}[\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]]} \approx \mathcal{C}_\phi \sqrt{\mathrm{V}^K[\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]]}, \tag{30}$$

with K sufficiently large.

3.3 On the generation of initial fields

Ensemble averaging benefits from independent initial conditions to generate uncorrelated flow evolutions. It is known that different turbulent flows will diverge with a rate determined by the Lyapunov exponent [46, 48], and that this is the case of our target problems. We decide then to generate perturbed initial conditions, and to let the system evolve for a defined burn-in time T_0 to arrive at uncorrelated solutions. We consider two strategies to generate these initial conditions.

The first approach simply consists of adding Gaussian white noise to a precomputed average velocity field $\bar{\boldsymbol{u}}$. In this work, this strategy of generating *spatially-uncorrelated* fluctuations is referred to as the spatially uncorrelated (SU) approach. We note that it is similar to the approach used in [40].

The second approach consists of adding nonlocal *spatially-correlated* and *divergence-free* solenoidal noise to the averaged velocity field $\bar{\boldsymbol{u}}$; we refer to this as the spatially correlated (SC) approach. Exploiting solenoidal fluctuations in the initial conditions is not new; we refer for example to [37], where the author used uncorrelated divergence-free initial conditions to ensure independence of different realizations. Our novelty is that we propose to generate spatially-correlated fluctuations $\boldsymbol{w}(\boldsymbol{x})$, which arise from a well-established synthetic turbulence model.

Our approach is inspired by the work of Hunt in [28] (see also [29, 34, 47]). The underlying assumption is that the inhomogenous contributions to fully developed turbulence fluctuations in the inviscid source layer above a solid body have negligible vorticity. From this assumption, one arrives at the following inhomogeneous turbulent fluctuation model: $\boldsymbol{w}(\boldsymbol{x}) = \boldsymbol{w}^{(\mathrm{H})}(\boldsymbol{x}) - \nabla\phi(\boldsymbol{x})$, where $\boldsymbol{w}^{(\mathrm{H})}(\boldsymbol{x})$ is a homogeneous turbulent velocity field and $\phi(\boldsymbol{x})$ satisfies

$$\Delta\phi = \nabla \cdot \boldsymbol{w}^{(\mathrm{H})} \quad \text{in } D, \quad (\nabla\phi - \boldsymbol{w}^{(\mathrm{H})}) \cdot \boldsymbol{n} = 0 \quad \text{on } \partial D. \tag{31}$$

In this work, we adopt the classical von Kárman model [55] for the homogeneous random field $\boldsymbol{w}^{(\mathrm{H})}(\boldsymbol{x})$. Realizations of this type of nonlocal spatially-correlated field can be generated using a Fourier transform on a Cartesian grid containing D; see, e.g., [41]. Once a realization $\boldsymbol{w}^{(\mathrm{H})}(\boldsymbol{x})$ is generated, we may interpolate the boundary conditions so that the solution to equation (31) can be solved with the same finite element spaces used in equation (5). After interpolating the sum $\boldsymbol{w}^{(\mathrm{H})}(\boldsymbol{x}) - \nabla\phi$, we arrive at the nonlocal spatially-correlated perturbation $\boldsymbol{w}(\boldsymbol{x})$ and, in turn, the SC initial condition $\boldsymbol{u}_0(\boldsymbol{x}) = \bar{\boldsymbol{u}}(\boldsymbol{x}) + \boldsymbol{w}(\boldsymbol{x})$.

3.4 On the optimal choice of the burn-in time

Given the full time interval $[0, T]$, we split it into a burn-in time interval $[0, T_0]$ and an effective time interval $[T_0, T]$. In this subsection, we focus on how to optimally choose T_0.

First, a single simulation is executed for a time long enough to reach a statistically stationary turbulent state, which is saved. Thereafter, N realizations are run with SU or SC initial conditions to ensure independent flow evolution. Once the required transient time T_0 is passed, statistical data are collected and updated on the fly, until the end of the effective time window.

We propose a systematic manner to minimize T_0, which makes use of the SE defined above. Given N realizations and a quantity of interest Q, our idea is to analyze how the statistical estimates of the quantity of interest (QoI) change for different burn-in times. We can observe this plotting the mean $\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]$ as function of T_0, together with its confidence intervals. The time interval $T - T_0$ is kept constant, while varying T_0. The confidence intervals are computed as $\mathcal{C}_\phi$SE, with confidence $1 - \phi$. By looking at the plot, we can detect a starting point after which the statistical result is effectively insensitive to T_0 variations. In addition to the statistical checks, we decide to apply a physical constraint, which in our case is the time the flow needs to go from the inlet to the obstacle. Therefore, T_0 will be the *maximum* of these two time values. In order to further reduce the computational cost of the transient phase, we also explore the possibility of using larger time steps in $[0, T_0]$.

Another way to estimate T_0 is analyzed in [6], where the authors choose a burn-in time which minimizes the estimated variance of the sample average estimator of the time average for a given signal. To do so, we average at each time step over all realizations, for different numbers of realizations N, and apply the procedure to the resultant time signal. As we will see in section 4, both procedures give similar results.

4 Numerical experiments

The first problem we consider (section 4.1) is the incompressible flow around a two-dimensional rectangle. We first check that statistical results are independent from the initial condition strategies, and we compare ensemble average against standard time averaging. Then, we check if it is possible to exploit a larger Courant-Friedrichs-Lewy (CFL) during the burn-in time phase, and how much the burn-in time window can be reduced. Finally, a comparison study between different strategies is made.

The second problem is presented in section 4.2 and describes wind flowing around a three-dimensional building. A comparison between ensemble averaging and standard time averaging is presented, together with the burn-in time study.

In both cases, a comparison against literature is made, to verify the implementation. We remark that international units are used to measure physical quantities. The computational efficiency of the joint use of XMC, Kratos and PyCOMPSs has already been demonstrated, where optimal strong scalabilty was ensured up to 128 nodes (6144 central processing units (CPUs)) [54]. The analyses were run on MareNostrum 4. This supercomputer has 11.15 Petaflops of peak performance, which consists of 3456 compute nodes equipped with two Intel R Xeon Platinum 8160 (24 cores at 2.1 GHz each) processors.

4.1 Rectangle body

4.1.1 Problem description

We focus on the flow around a $5 \times 1\,\mathrm{m}$ rectangular body [8], characterized by a Reynolds number of 132719. The problem domain is shown in figure 4.1.

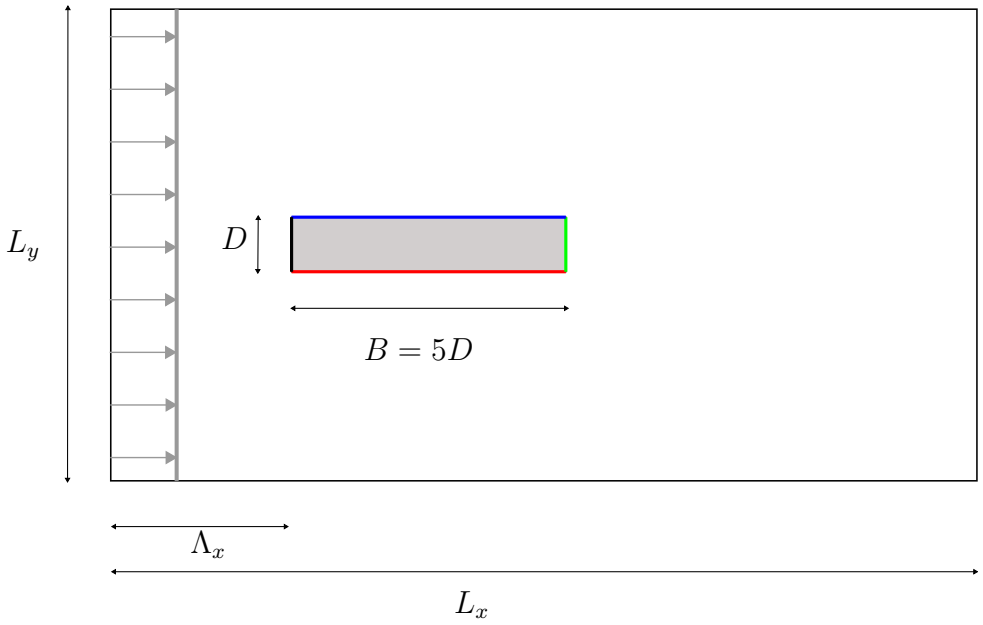


Figure 4.1: Scheme of the computational domain used for the rectangle problem, where $D = 1\,\mathrm{m}$, $B = 5D$, $L_x = 55B$, $L_y = 30B$ and $\Lambda_x = 15B$. Thus, the dimensions of the outer domain are $275 \times 150\,\mathrm{m}$, and the inner rectangle has size $5 \times 1\,\mathrm{m}$.

The governing equations of the problem are defined in section 2. Slip boundary conditions are enforced on the external boundaries, and no-slip boundary conditions on the rectangle body. The inlet velocity is uniformly distributed on the y-axis, and it has a value of $2\,\mathrm{m\,s^{-1}}$.

The mesh considered to solve the problem has around 25000 nodes, and a minimal size, close to the rectangle body, of 0.002 m. The chosen time step is 0.02 s, which gives a CFL of 20. Such mesh is adaptive with respect to a solution-oriented metric, namely the average velocity field $\langle \boldsymbol{u}(t,x) \rangle_{T_0,T}$. The metric is computed exploiting Kratos [43], and the original mesh is refined using the Mmg software [18].

The quantities of interest are the drag force F_d on the body, the pitching moment M_p on the body and the pressure field $p(x)$ on all nodes of the body surface. The international system of units is used for all the results. However, even though we compute all these

quantities of interest, we assess statistical convergence only for the drag force. For the sake of simplicity, we may use $\langle \mathrm{Q} \rangle$ instead of $\langle \mathrm{Q} \rangle_{T_0,T}$.

4.1.2 Perturbation of initial conditions

We can observe in figure 4.2 the time evolution of the time averaged drag force $\langle F_d \rangle_{40\,\mathrm{s},t}$ for 128 contributions, when using the correlated and divergence-free initial condition strategy. The burn-in time we select is 40 s, which is the optimal T_0 we find in section 4.1.4. Similar plots are obtained for SU initial conditions and for others T_0.

Each of these samples runs for 600 s, which, as we can observe by looking at the oscillations we have, is a time horizon not long enough to reach convergence for the time averaged drag value. In case of infinitely large time windows, one would expect each realization to converge to the same value. Since this is not feasible, figure 4.2 gives us an estimate of the error that is being committed by considering truncated time windows. The estimations of expected value, standard deviation and statistical error for both perturbations are reported in table 1.

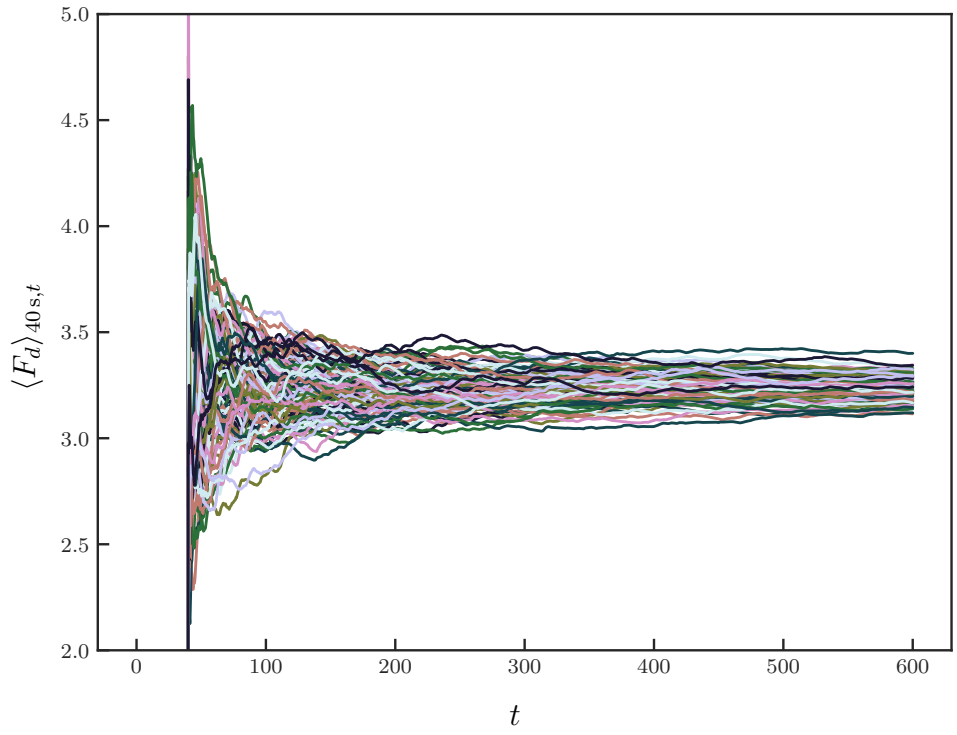


Figure 4.2: Time averaged drag force $\langle F_d \rangle_{40\,\mathrm{s},t}$ evolution as function of time. Initial conditions are perturbed following the SC initial conditions.

From the drag force of table 1, one can estimate the drag coefficient as

$$C_d = \frac{F_d}{\frac{1}{2}\rho \boldsymbol{u}^2 A}, \tag{32}$$

where ρ is the fluid density, $\boldsymbol{u}$ the speed of the rectangle body relative to the fluid and A the cross sectional area. The drag coefficient we obtain is $C_d = 1.321$, which is consistent with literature results [8].

	$\mathrm{E}^N[\mathrm{Q}]$	$\sigma^N[\mathrm{Q}]$	$\mathcal{C}_\phi \mathrm{SE}$
SU	3.238648	0.052823	0.010819
SC	3.246658	0.055877	0.011444

Table 1: Estimations of expected value, standard deviation and statistical error with 99% confidence for SU and SC initial conditions. The QoI Q is the time averaged drag force $\langle F_d \rangle_{40\,\mathrm{s},600\,\mathrm{s}}$. 128 realizations are considered.

4.1.3 Comparison ensemble average and time average

In order to compare ensemble averaging and standard time averaging approaches, we compute and compare the total error, given by the left hand side of equation (27), for different computational costs.

First, we analyze which are the dominant terms of equation (27). We plot $\left(\mathrm{V}^K[\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]]\right)^{-1}$ for $\mathrm{Q} = F_d$ and different K and N in figure 4.3. In the plot, black dots are estimations of the reciprocal of the variance, while the red line is the linear interpolation of such estimations. We observe that the variance estimation decays linearly as N and $(T - T_0)$ grow. Therefore, the fast decay of section 3.2 is happening and the dominant term of the total error is $\mathcal{C}_\phi \sqrt{\mathrm{V}^K[\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]]}$. We can then simplify equation (27) to $\mathcal{C}_\phi \mathrm{SE} \approx \mathcal{C}_\phi \sqrt{\mathrm{V}^K[\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]]} \leq \varepsilon$.

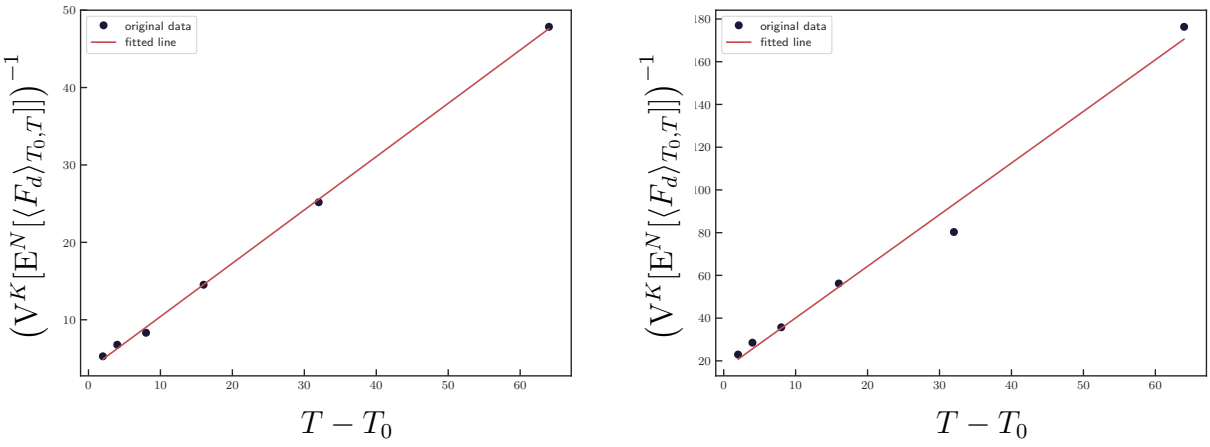


Figure 4.3: Computation of $\mathrm{V}^K[\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]]$ as function of $T - T_0$ for $\mathrm{Q} = F_d$. The left plot presents $(K, N, T_0) = (128, 1, 40\,\mathrm{s})$ and the right plot $(K, N, T_0) = (32, 4, 40\,\mathrm{s})$.

Tables 2 and 3 show the SE for both ensemble average and standard time average approaches. First, we observe SE decreases as expected (proportional to $N^{-0.5}$ and $(T - T_0)^{-0.5}$) as more realizations or larger time windows are considered. For example, let's focus on the second line of table 2 and the sixth of table 3. SE values are approximately similar, but ensemble average employs 10 samples that can be run concurrently. Consequently, the total computational cost corresponds to running 10 simulations, each with an effective time window of 160 s. Moreover, we point out that the expected value estimations of the two average strategies are in agreement within each other. As expected, ensemble average approach drastically reduces the time to solution, *for the same statistical error*, provided of course that more computing resources are used to enable the concurrent solution of the ensemble. This means that more working nodes are used as

$E^N[\langle F_d \rangle]$	SE	N	$T - T_0$	C	time to solution
3.2779	0.0403	5	160	136	3.40
3.2684	0.0285	10	160	272	3.40
3.2583	0.0201	20	160	544	3.40
3.2527	0.0142	40	160	1088	3.40
3.2584	0.0100	80	160	2176	3.40
3.2456	0.0071	160	160	4352	3.40

Table 2: The table reports the mean estimation and its associated SE computed for ensemble averaging for the drag force mean estimation. N and $T - T_0$ refer to the number of realizations and the effective time window of the simulation, respectively. C is the computational cost and is expressed in CPU hours. Time to solution is the human time we need to wait for getting results, and is expressed in hours.

$E^N[\langle F_d \rangle]$	SE	N	$T - T_0$	C	time to solution
3.3056	0.1595	1	50	17.2	2.15
3.2171	0.1127	1	100	21.76	2.72
3.2752	0.0797	1	200	30.8	3.85
3.2605	0.0563	1	400	48.96	6.12
3.2472	0.0398	1	800	85.28	10.66
3.2552	0.0281	1	1600	158.02	19.75

Table 3: The table reports the mean estimation and its associated SE computed for standard time averaging for the drag force mean estimation. N and $T - T_0$ refer to the number of realizations and the effective time window of the simulation, respectively. C is the computational cost and is expressed in CPU hours. Time to solution is the human time we need to wait for getting results, and is expressed in hours.

more realizations are run. If enough resources are allocated, the runtime is shorter, and this is our case.

The results suggest that the ensemble average approach is more appropriate than standard time averaging for running in supercomputers, since it allows to fully exploit supercomputer capabilities in order to reduce the time to solution. This comes at the price of a larger combined computational cost, due to the need of going multiple times through the initial burn-in time.

4.1.4 On the reduction of burn-in time computational cost

We analyze how the statistical results of the time-averaged drag force change when varying the burn-in time. For each case we consider 128 realizations, and we keep constant $T - T_0$. We plot in figure 4.4 the expected value estimation as function of T_0, together with its 99% confidence intervals. We observe that the statistical result is relatively insensitive to T_0 for $T_0 > 20\,\mathrm{s}$.

As mentioned in section 3, another way to estimate T_0 is following the approach presented in [6], in which the authors choose a burn-in time which minimizes the estimated

variance of the sample average estimator of the time average for a given signal. Figure 4.5 reports the estimated variance of the sample average estimator of the time average as function of the burn-in time, for different number of realizations and fixed $T - T_0$. As we can see, we reach the minimum after few seconds.

Even though both way of estimating the burn-in time suggest that only a very short time span is needed, we consider as "physical constraint" the time required by the information to travel from the inlet to past the object. This means that we wait at least the physical constraint time before we can start trusting the solver results. The time needed for this to happen, for an average speed of 2 m/s, is 40 s. Our conclusion is that we can safely assume $T_0 = 40\,\mathrm{s}$ without changing statistical results. We remark as well that the same conclusion follow for other effective time windows $T - T_0$.

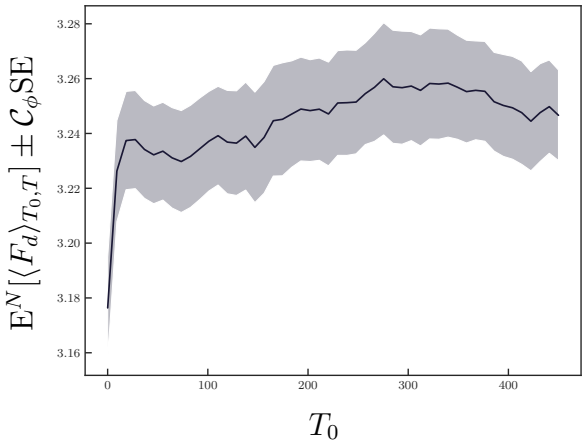


Figure 4.4: Expected value estimation and associated SE for a confidence of 99% as a function of the burn-in time. $N = 128$ and $T - T_0 = 150\,\mathrm{s}$.

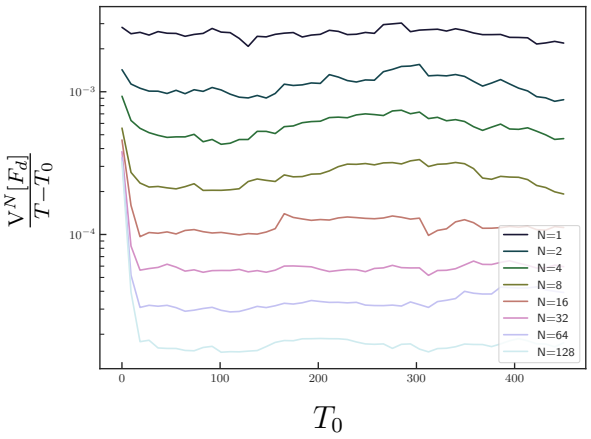


Figure 4.5: Ratio between drag force variance and effective time window, for different realizations and $T - T_0 = 150\,\mathrm{s}$.

Another approach we consider to reduce the time to solution is to *exploit larger time steps* during the burn-in time. To do so, we must first verify that a larger time step does not change the statistical results. Moreover, to improve the consistency of such an approach, we should ensure that the chosen larger time step, if used during the whole time window, would give different results.

We report in table 4 the expected value estimation and the associated SE for a confidence of 99%. We can see that in the case of time time step $\Delta t = 0.02$ used in the averaging window $[T_0, T]$, we obtain consistent statistical results, independently from Δt_0 values. On the other hand, for a different Δt we obtain a different statistical result (see third row in the table compared to the first two). Therefore, a larger time step can safely be employed to reduce the time to solution of the burn-in phase.

4.1.5 Results

Combining all of the ideas presented above we obtain the results reported in table 5, which shows the statistical analysis of the time averaged drag force. The analyses presented in table 5 are driven by constant product between $N = 128$ and $T = 300\,\mathrm{s}$. Convergence is checked via equation (27), which is simplified to $\mathcal{C}_\phi \mathrm{SE} \leq \varepsilon$. The absolute tolerance is $\varepsilon = 0.02$ (the relative value is $\approx 0.6\%$) and the confidence is $1 - \phi = 0.99$.

$\mathrm{E}^N[\langle F_d\rangle_{T_0,T}]$	$\mathcal{C}_\phi$SE	$T-T_0$	Δt	Δt_0
3.237284	0.009183	760	0.02	0.02
3.232869	0.008286	760	0.02	0.05
3.312648	0.008199	760	0.05	0.05

Table 4: Expected value estimation and associated SE with 99% confidence for different time steps during both burn-in and effective phases.

Different T_0, Δt_0 and perturbation of initial conditions are considered. On one hand, we consider the optimal $T_0 = 40\,\mathrm{s}$, on the other $T_0 = 140\,\mathrm{s}$, which is directly related to the time one particle needs to travel from the inlet to the outlet for an average speed of $2\,\mathrm{m\,s^{-1}}$. We observe all strategy gives the same statistical result, since all expected value estimations fall within the range of confidence $1-\phi = 0.99$. Moreover, both the time to solution and the computational cost are smaller if larger time steps are exploited in the burn-in time phase. Therefore, we conclude the most promising strategy consists in exploiting $\frac{\Delta t_0}{\Delta t} > 1$ and T_0 small enough but still ensuring the error decays as $N^{-0.5}$ and $(T-T_0)^{-0.5}$.

$\mathrm{E}^N[\langle F_d\rangle_{T_0,T}]$	$\mathcal{C}_\phi$SE	initial conditions	N	$T-T_0$	T_0	$\frac{\Delta t_0}{\Delta t}$	C	time to solution
3.245649	0.018503	SU	128	160	140	1.0	3655	3.57
3.244356	0.013949	SU	128	260	40	1.0	3655	3.57
3.237846	0.014395	SU	128	260	40	2.5	3389	3.31
3.235604	0.019199	SC	128	160	140	1.0	3727	3.64
3.235677	0.014501	SC	128	260	40	1.0	3727	3.64
3.236612	0.014390	SC	128	260	40	2.5	3420	3.34

Table 5: Statistical analyses of time averaged drag force $\langle F_d\rangle_{T_0,T}$. The expected value estimation and the associated statistical error, with a confidence $1-\phi = 0.99$, are reported. Both uncorrelated and correlated initial condition perturbations are presented. N refers to the number of ensembles realizations. Effective time window $T-T_0$ and burn-in time T_0 are expressed in seconds, and $\frac{\Delta t_0}{\Delta t}$ shows if a larger CFL is used in the transient phase. The computational cost C and time to solution unit measures are CPU hours and hours, respectively. The product between number of realizations and time window is constant among different analyses.

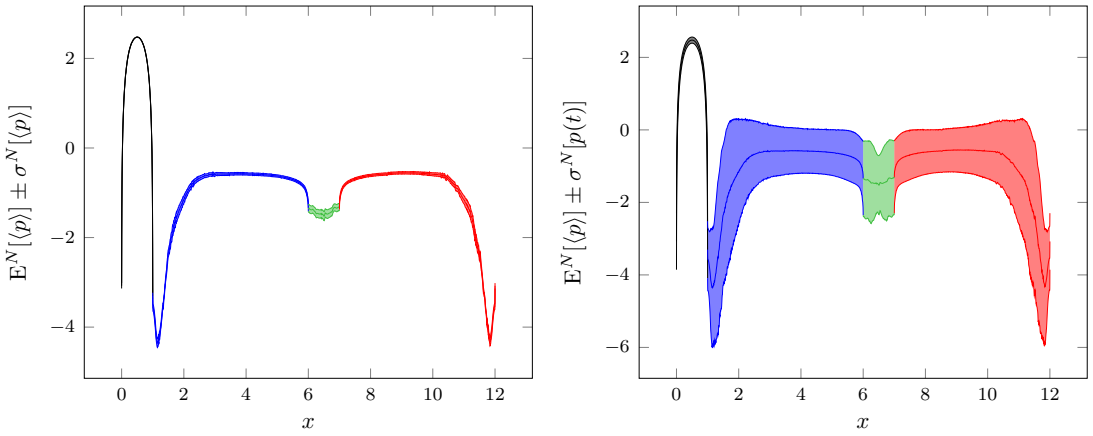
4.1.6 Other observables

In addition to the statistical analysis reported above for the drag force, we present here results for the expected value and standard deviation estimations of the drag force, pitching moment and pressure field on the rectangle body. Specifically, we compute the standard deviation of an observable Q and of its time average, which read $\sigma[\mathrm{Q}]$ and $\sigma[\langle \mathrm{Q}\rangle_{T_0,T}]$, respectively. $\sigma[\mathrm{Q}]$ can be understood as an indicator of the distribution around the mean value, while $\sigma[\langle \mathrm{Q}\rangle_{T_0,T}]$ as an error indicator of the expected value estimation.

Table 6 reports results for the drag force and the base moment, while figures 4.6 (a) and 4.6 (b) for the pressure field.

Q	$\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]$	$\sigma^N[\langle \mathrm{Q} \rangle_{T_0,T}]$	$\sigma^N[\mathrm{Q}]$
F_d	3.238950	0.086650	0.575146
M_p	-0.014169	0.123063	2.141448

Table 6: Statistical analysis of the drag force F_d and of the pitching moment M_p.



(a) Estimation of the expected value and of the time-averaged standard deviation of the pressure field $p(\boldsymbol{x})$ around the rectangle body. Each color represent one side of the rectangle, and the color of each side is presented in figure 4.1.

(b) Estimation of the expected value and of the standard deviation of the pressure field $p(\boldsymbol{x})$ around the rectangle body. Each color represent one side of the rectangle, and the color of each side is presented in figure 4.1.

Figure 4.6: Risk measures for the rectangle problem.

4.2 CAARC building

4.2.1 Problem description

The second problem is the wind flow around the *Commonwealth Advisory Aeronautical Council* (CAARC) building [7, 24, 25, 49]. The CAARC is a parallelepiped building with width 45 m, length 30 m and height 180 m. The domain is 1880 m long, 864 m large and 576 m high. A steady state logarithmic wind profile is considered. The wind mean profile is described by [30]

$$\bar{\boldsymbol{u}}(z) = \frac{\boldsymbol{u}^*}{k} \ln(z/z_0), \tag{33}$$

where $k \approx 0.4$ is von Karman's constant, $\boldsymbol{u}^*$ the friction velocity and z_0 the roughness length. The reference mean wind velocity $\bar{\boldsymbol{u}}(z)$ is defined at reference height H.

In table 7 we present the physical properties of the problem, and we remark that $z_0 = 2\,\mathrm{m}$ is typical of centers of large cities [30]. The Reynolds number is 119 millions, where a characteristic length of 45 m is considered. Under such conditions, it is clear that the problem is badly under-resolved.

$\bar{\boldsymbol{u}}_H$	H	z_0	ρ	μ	Re
$40\,\mathrm{m\,s^{-1}}$	180 m	2 m	$1.225\,\mathrm{kg/m^3}$	$1.846 \cdot 10^{-5}\,\mathrm{kg\,m^{-1}\,s^{-1}}$	119447453

Table 7: Physical parameters problem.

The quantities of interest are the drag force F_d on the body, the base moment M_b on the body and the pressure field $p(x)$ on all nodes of the body surface. Units are in the SI system. The quantity of interest we choose to analyze is the drag force. As above, we omit the subscript T_0, T if there is no risk of misunderstanding.

The mesh considered to solve the problem has approximately 312000 nodes, and a minimal size, close to the body, of 0.2 m. The considered mesh is adaptive with respect to a metric built on top of velocity and pressure fields.

4.2.2 Validation

We compute and compare the normalized formulas for forces and moments against [7]. The formulas read

$$\begin{aligned} C_{F_X} &= \frac{F_X}{1/2\rho W \int_0^H \bar{u}^2 \,\mathrm{d}Z}, \quad C_{F_Y} = \frac{F_Y}{1/2\rho W \int_0^H \bar{u}^2 \,\mathrm{d}Z}, \\ C_{M_X} &= \frac{M_X}{1/2\rho \bar{u}_H^2 W H^2}, \quad C_{M_Y} = \frac{M_Y}{1/2\rho \bar{u}_H^2 W H^2}, \end{aligned} \tag{34}$$

where ρ is the density of the fluid, $\bar{\boldsymbol{u}}$ the velocity mean profile described in equation 33, $\bar{\boldsymbol{u}}_H$ the velocity at height $H = 180$ m, and W the building width. C_{F_X} and C_{F_Y} represent the force coefficients in the direction X and Y, respectively. M_{F_X} and M_{F_Y} denote the moment coefficients in the same directions, where the moment is computed around the centroid of the plan geometry of the building at ground location. Figure 4.7 shows a good

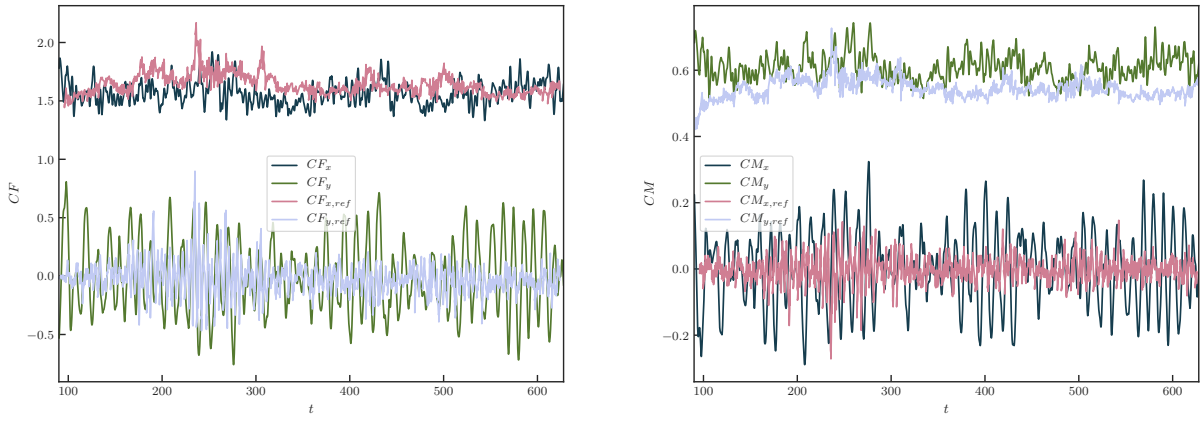


Figure 4.7: Drag and moment coefficients comparison between our work and [7].

agreement of our solution with respect to literature.

4.2.3 Comparison ensemble average and time average

We have observed in section 4.1 that perturbing initial conditions with SU or SC noise is equivalent, from both computational and statistical points of view. For this reason, we prefer to use the latter, since more consistent from a physical point of view.

First, we analyze in figure 4.8 which are the dominant terms of equation (27) by plotting $\left(\mathrm{V}^K[\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]]\right)^{-1}$ for different K and N and $\mathrm{Q} = F_d$. The linear decay of the variance estimation with respect to N and $(T - T_0)$ suggests that the dominant term of the total error is $\mathcal{C}_\phi \sqrt{\mathrm{V}^K[\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]]}$, which implies simplifying equation (27) to $\mathcal{C}_\phi \mathrm{SE} \approx \mathcal{C}_\phi \sqrt{\mathrm{V}^K[\mathrm{E}^N[\langle \mathrm{Q} \rangle_{T_0,T}]]} \leq \varepsilon$.

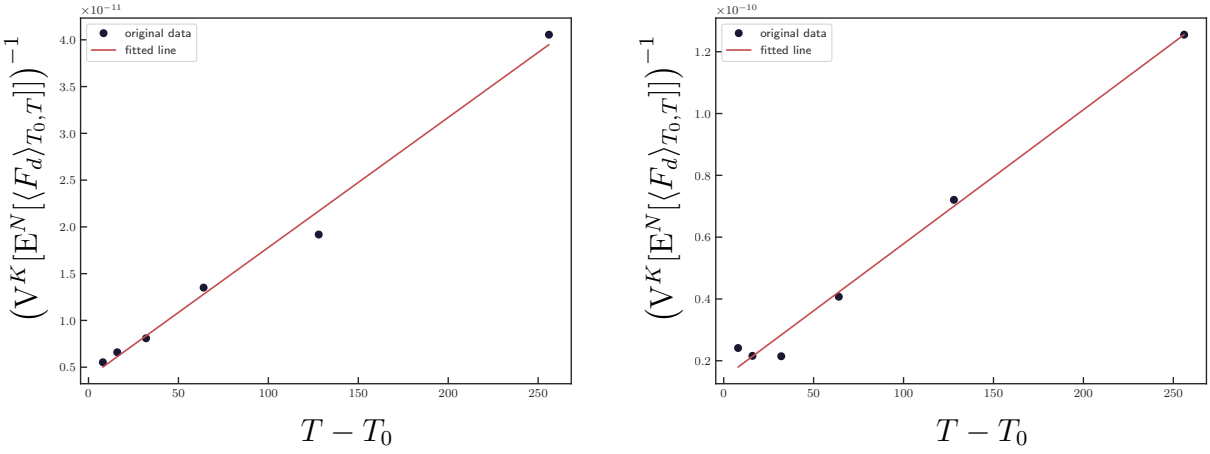


Figure 4.8: Computation of $\mathrm{V}^K[\mathrm{E}^N[\langle \mathrm{Q}\rangle_{T_0,T}]]$ as function of $T - T_0$ for $\mathrm{Q} = F_d$. The left plot presents $(K, N, T_0) = (128, 1, 30\,\mathrm{s})$ and the right plot $(K, N, T_0) = (32, 4, 30\,\mathrm{s})$.

We compute the SE for ensemble averaging and standard time averaging in tables 8 and 9. The SE decreases as expected as long as more realizations or larger time windows are considered. Moreover, the ensemble average approach drastically reduces the time to solution, for the same statistical error. For example, the case $N = 4$, $T - T_0 = 210\,\mathrm{s}$ of the ensemble average approach, compared to $N = 1$, $T - T_0 = 840\,\mathrm{s}$ of standard time average, reduces the time to solution by almost a factor 4, to obtain a similar SE. We remark as well that the expected value estimations for both ensemble averaging and standard time averaging are consistent within each other.

$\mathrm{E}^N[\langle F_d\rangle]$	SE	N	ΔT	C	time to solution
8982493	84767	4	210	1666	17.36
8932223	59939	8	210	3333	17.36
8973444	42383	16	210	6666	17.36
8986913	29969	32	210	13332	17.36
8927955	21191	64	210	26664	17.36
8930547	14984	128	210	53329	17.36

Table 8: The table reports the mean estimation and its associated SE of ensemble averaging for the estimation of the drag force mean. N and $\Delta T = T - T_0$ refer to the number of realizations and the effective time window of the simulation, respectively. C is the computational cost and is expressed in CPU hours, while time to solution is expressed in hours.

4.2.4 On the reduction of burn-in time computational cost

We analyze now if it is statistically consistent to reduce the burn-in time. As before, we consider 128 realizations and we keep constant $T - T_0 = 110\,\mathrm{s}$. By looking at figures 4.9 and 4.10, we conclude that the burn-in time can be reduced to any value larger than 30 s. The same conclusion holds also for different effective time windows. To ensure robustness

$\mathrm{E}^N[\langle F_d \rangle]$	SE	N	ΔT	C	time to solution
8658884	324256	1	52.5	215.58	8.98
8879404	229284	1	105	300.19	12.50
9109719	162128	1	210	468.56	19.52
9003445	114642	1	420	853.28	35.55
8950303	93604	1	630	1189.42	49.55
8956216	81064	1	840	1524.88	63.53

Table 9: The table reports the mean estimation and its associated SE of standard time averaging for the estimation of the drag force mean. N and $\Delta T = T - T_0$ refer to the number of realizations and the effective time window of the simulation, respectively. C is the computational cost and is expressed in CPU hours, while time to solution is expressed in hours.

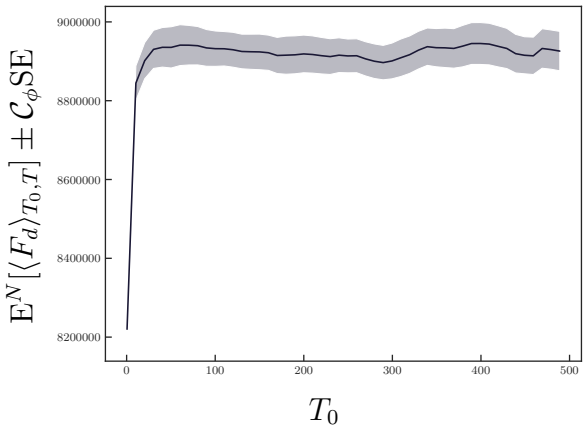


Figure 4.9: Expected value estimation and associated SE for a confidence of 99% as a function of the burn-in time. $N = 128$ and $T - T_0 = 110\,\mathrm{s}$.

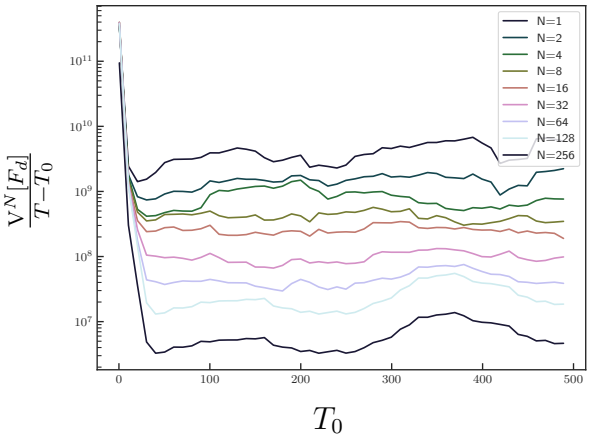


Figure 4.10: Ratio between drag force variance and effective time window, for different realizations and $T - T_0 = 110\,\mathrm{s}$.

of our strategy, we apply the physical constraint that T_0 should be larger than the time to travel from the inlet to the body. Since for an average velocity of $40\,\mathrm{m\,s^{-1}}$ such time is $11.625\,\mathrm{s}$, it is statistically consistent to use $T_0 = 30\,\mathrm{s}$.

We also check if it is possible to exploit larger time steps during the burn-in time, in order to reduce its computational cost. Table 10 shows that running with a larger time step during T_0 is statistically equivalent to exploit a constant time step, where $\mathcal{C}_\phi$ is computed for a 99% confidence.

$\mathrm{E}^N[\langle F_d \rangle_{T_0,T}]$	$\mathcal{C}_\phi$SE	$T - T_0$	$\frac{\Delta t_0}{\Delta t}$
8922399	39013	170	1.0
8907406	39534	170	2.5

Table 10: Expected value and associated SE with 99% confidence for different time steps during burn-in time.

4.2.5 Results

Finally, we run the problem exploiting ensemble average, larger time step Δt_0 and smaller burn-in time $T_0 = 30\,\text{s}$. Convergence is checked with equation (27), which is simplified to $\mathcal{C}_\phi \text{SE} \leq \varepsilon$. The chosen confidence is 99%, and the relative tolerance with respect to the time averaged drag force mean estimator is around 0.5%. We run the problem for different configurations, keeping the overall cost given by the product between time window and number of realizations $T \cdot N$ approximately constant. Results are shown in table 11.

$\text{E}^N[\langle F_d \rangle_{T_0,T}]$	$\mathcal{C}_\phi \text{SE}$	N	$T - T_0$	T_0	$\frac{\Delta t_0}{\Delta t}$	C	time to solution
8972727	48924	142	110	30	2.5	35749	10.34
8946768	45797	100	170	30	2.5	38041	15.54
8943515	40755	76	230	30	2.5	38566	20.60

Table 11: The table reports the expected value and the statistical error values of the time averaged drag force $\langle F_d \rangle_{T_0,T}$, with a 99% confidence. N, T and T_0 refer to the number of ensemble realizations, the time window $[0, T]$ upper bound of the simulation and the burn-in time, respectively. These last two are measured in seconds. $\frac{\Delta t_0}{\Delta t}$ is the ratio between the time steps of T_0 and of the effective time window $T - T_0$. C is the computational cost, expressed in CPU hours, and time to solution is the real time we need to wait for the solution and is expressed in hours.

4.2.6 Other observables

We select the case with minimal statistical error of table 11 to show the statistical results for other quantities of interest. Table 12 shows the expected value and the standard deviation estimators for the drag force and the base moment. Figures 4.11 and 4.12 show the estimations of expected value and standard deviation for the pressure field.

Q	$\text{E}^N[\langle \text{Q} \rangle_{T_0,T}]$	$\sigma^N[\langle \text{Q} \rangle_{T_0 T}]$	$\sigma^N[\text{Q}]$
F_d	8943515	152726	662129
M_b	-14943	436540	7332992

Table 12: Statistical analysis of drag force and base moment.

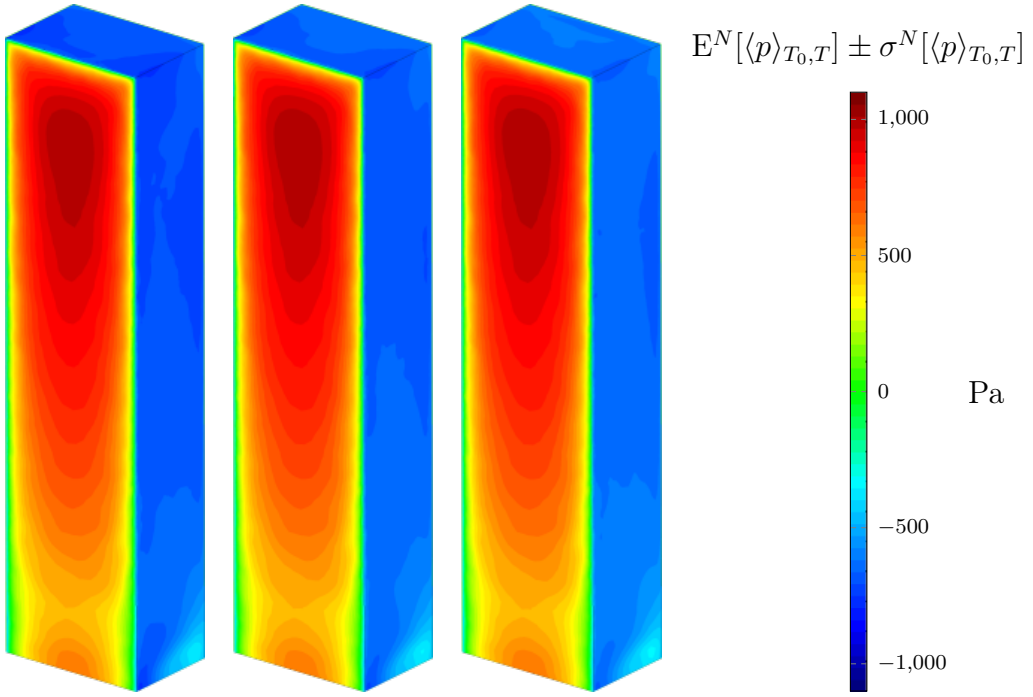


Figure 4.11: Statistical result of the pressure field $\langle p(x)\rangle_{T_0,T}$. From left to right, $\mathrm{E}^N[\langle p(x)\rangle_{T_0,T}] - \sigma^N[\langle p(x)\rangle_{T_0,T}]$, $\mathrm{E}^N[\langle p(x)\rangle_{T_0,T}]$ and $\mathrm{E}^N[\langle p(x)\rangle_{T_0,T}] + \sigma^N[\langle p(x)\rangle_{T_0,T}]$.

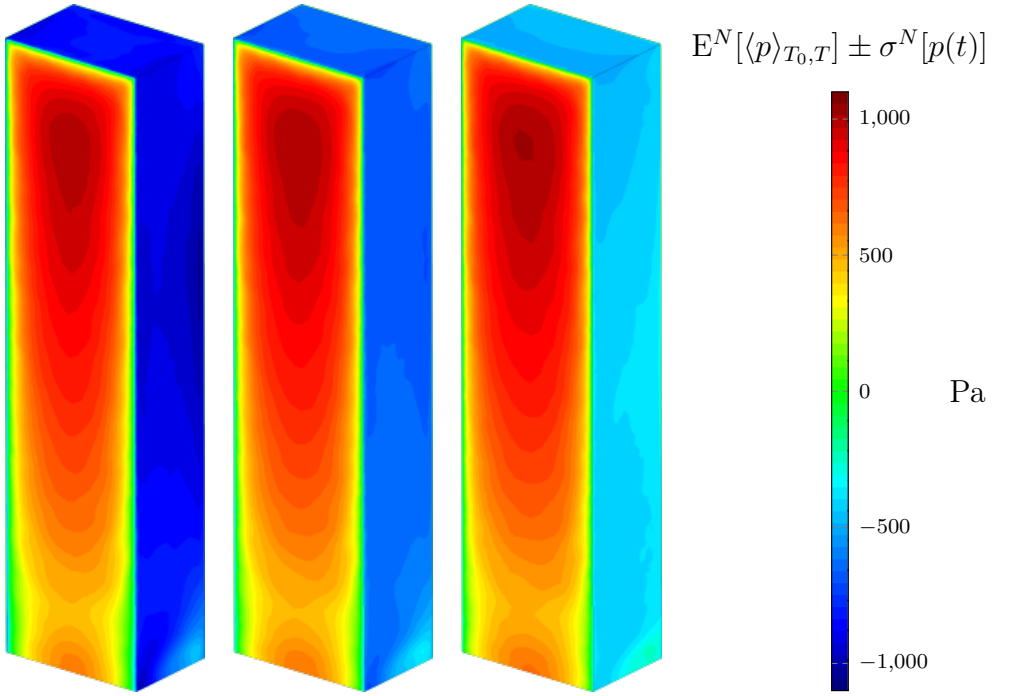


Figure 4.12: Statistical result of the pressure field $p(x)$. From left to right, $\mathrm{E}^N[\langle p(x)\rangle_{T_0,T}] - \sigma^N[p(x)]$, $\mathrm{E}^N[\langle p(x)\rangle_{T_0,T}]$ and $\mathrm{E}^N[\langle p(x)\rangle_{T_0,T}] + \sigma^N[p(x)]$.

5 Software release

In addition to this report, deliverable 3.3 comprises a new version of the Kratos Multiphysics library. This version 9.0 of the Kratos Multiphysics library has been publicly re-

leased online [42]. Different examples showing how to apply the ensemble average method to solve computational fluid dynamics problems can be found online [44].

6 Conclusions

In this work, we show that ensemble averaging can be successfully applied to highly chaotic incompressible flows and we propose strategies to minimize the total error and the computational cost of the simulation. Two numerical examples are considered to demonstrate the advantage of using ensemble averaging over standard time averaging when running in HPC systems and to validate our proposals.

The statistical analysis of ensemble averaging expected value estimator leads to the identification of two error components: an initialization bias, related to the transient perturbation of the flow, and a statistical error, related to finite sampling. Convergence rates of both error contributions are analyzed by considering two scenarios: one with fast decay of the transient perturbation and one with slow decay. This allows understanding how the error contributions should decay in order to assume null initialization bias. For both numerical examples, decay rates are estimated to assess if the initialization bias is negligible. For both problems, the burn-in phase computational cost is minimized by following a statistical-based approach and a less accurate and less expensive time integration procedure during the burn-in phase.

Multiple observables (drag force, base and pitching moment and pressure field) are computed within this work. By applying the proposed statistical ensemble averaging framework, statistical estimators are efficiently and accurately estimated, and decisions based on top of such statistics can therefore be taken faster.

References

[1] J. Ang, K. Evans, A. Geist, M. Heroux, P. Hovland, O. Marques, L. C. McInnes, E. Ng, and S. Wild. Report on the workshop on extreme-scale solvers: Transitions to future architectures. Technical report, U.S. Department of Energy, 2012.

[2] Q. Ayoul-Guilmard, S. Ganesh, F. Nobile, R. Rossi, R. Tosi, R. M. Badia, and R. Amela. XMC, 2020.

[3] R. M. Badia, J. Conejero, C. Diaz, J. Ejarque, D. Lezzi, F. Lordan, C. Ramon-Cortes, and R. Sirvent. COMP Superscalar, an interoperable programming framework. *SoftwareX*, 3–4, 2015.

[4] S. Badia and J. V. Gutiérrez-Santacreu. Convergence towards weak solutions of the Navier-Stokes equations for a finite element approximation with numerical subgrid scale modeling. *IMA Journal of Numerical Analysis*, In press, 2013.

[5] J. Banks, J. Carson, B. L. Nelson, and D. Nicol. *Discrete-Event System Simulation (4th Edition)*. Prentice Hall, 4 edition, 2004.

[6] P. Beyhaghi, S. Alimohammadi, and T. Bewley. Uncertainty Quantification of the time averaging of a Statistics Computed from Numerical Simulation of Turbulent Flow. feb 2018.

[7] A. L. Braun and A. M. Awruch. Aerodynamic and aeroelastic analyses on the CAARC standard tall building model using numerical simulation. *Computers and Structures*, 87(9-10):564–581, may 2009.

[8] L. Bruno, M. V. Salvetti, and F. Ricciardelli. Benchmark on the aerodynamics of a rectangular 5:1 cylinder: An overview after the first four years of activity. *Journal of Wind Engineering and Industrial Aerodynamics*, 126:87–106, mar 2014.

[9] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1):12–26, aug 1967.

[10] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22(104):745–745, 1968.

[11] R. Codina. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering*, 191(39-40):4295–4321, 2002.

[12] R. Codina, S. Badia, J. Baiges, and J. Principe. Variational Multiscale Methods in Computational Fluid Dynamics. In *Encyclopedia of Computational Mechanics Second Edition*, pages 1–28. John Wiley & Sons, Ltd, Chichester, UK, 2017.

[13] R. Codina, J. Principe, O. Guasch, and S. Badia. Time dependent subscales in the stabilized finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering*, 196(21-24):2413–2430, 2007.

[14] O. Colomés, S. Badia, R. Codina, and J. Principe. Assessment of variational multiscale models for the large eddy simulation of turbulent incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 285:32–63, mar 2015.

[15] O. Colomés, S. Badia, and J. Principe. Mixed finite element methods with convection stabilization for the large eddy simulation of incompressible turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 304(1):294–318, jun 2016.

[16] P. Dadvand, R. Rossi, M. Gil, X. Martorell, J. Cotela, E. Juanpere, S. R. Idelsohn, and E. Oñate. Migration of a generic multi-physics framework to HPC environments. *Computers and Fluids*, 80(1):301–309, 2013.

[17] P. Dadvand, R. Rossi, and E. Oñate. An object-oriented environment for developing finite element codes for multi-disciplinary applications. *Archives of Computational Methods in Engineering*, 17(3):253–297, 2010.

[18] C. Dapogny, C. Dobrzynski, and P. Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of Computational Physics*, 262:358–378, apr 2014.

[19] R. Durrett. *Probability: Theory and Examples*. 2019.

[20] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers : with Applications in Incompressible Fluid Dynamics: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2005.

[21] W. Fang and M. B. Giles. Multilevel Monte Carlo method for ergodic SDEs without contractivity. *Journal of Mathematical Analysis and Applications*, 2019.

[22] M. J. Gander. 50 Years of Time Parallel Time Integration. pages 69–113. Springer, Cham, 2015.

[23] F. F. Grinstein, L. G. Margolin, and W. J. Rider. *Implicit Large Eddy Simulation: Computing Turbulent Fluid Dynamics*. Cambridge University Press, july 2007.

[24] J. D. Holmes and T. K. Tse. International high-frequency base balance benchmark study. *Wind and Structures, An International Journal*, 18(4):457–471, 2014.

[25] S. Huang, Q. S. Li, and S. Xu. Numerical evaluation of wind effects on a tall steel building by CFD. *Journal of Constructional Steel Research*, 2007.

[26] T. J. R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, 127(1–4):387–401, nov 1995.

[27] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy. The variational multiscale method—a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1–2):3–24, nov 1998.

[28] J. C. Hunt. A theory of turbulent flow round two-dimensional bluff bodies. *Journal of Fluid Mechanics*, 61(4):625–706, 1973.

[29] J. C. Hunt. Turbulence structure in thermal convection and shear-free boundary layers. *Journal of Fluid Mechanics*, 138:161–184, 1984.

[30] J. JCSS. Probabilistic model code. *Joint Committee on Structural Safety*, 2001.

[31] E. W. Jenkins, V. John, A. Linke, and L. G. Rebholz. On the parameter choice in grad-div stabilization for the Stokes equations. *Advances in Computational Mathematics*, 40(2):491–516, sep 2013.

[32] N. Jiang. A Higher Order Ensemble Simulation Algorithm for Fluid Flows. *Journal of Scientific Computing*, 64(1):264–288, 2015.

[33] N. Jiang and W. Layton. An algorithm for fast calculation of flow ensembles. *International Journal for Uncertainty Quantification*, 4(4):273–301, 2014.

[34] B. Keith, U. Khristenko, and B. Wohlmuth. A fractional PDE model for turbulent velocity fields near solid walls. *Journal of Fluid Mechanics*, 916, 2021.

[35] D. E. Keyes. Exaflop/s Pourquoi et comment. *Comptes Rendus - Mecanique*, 339(2-3):70–77, 2011.

[36] B. Krasnopolsky, N. Nikitin, and A. Lukyanov. Optimizing generation of multiple turbulent flow states. *Journal of Physics: Conference Series*, 1129(1):12020, 2018.

[37] B. I. Krasnopolsky. An approach for accelerating incompressible turbulent flow simulations based on simultaneous modelling of multiple ensembles. *Computer Physics Communications*, 229:8–19, 2018.

[38] B. I. Krasnopolsky. Generation of multiple turbulent flow states for the simulations with ensemble averaging. *Supercomputing Frontiers and Innovations*, 5(2):55–62, 2018.

[39] F. Lordan, E. Tejedor, J. Ejarque, R. Rafanell, J. Álvarez, F. Marozzo, D. Lezzi, R. Sirvent, D. Talia, and R. M. Badia. ServiceSs: An Interoperable Programming Framework for the Cloud. *Journal of Grid Computing*, 12(1):67–91, 2014.

[40] V. Makarashvili, E. Merzari, A. Obabko, A. Siegel, and P. Fischer. A performance analysis of ensemble averaging for high fidelity turbulence simulations at the strong scaling limit. *Computer Physics Communications*, 219:236–245, 2017.

[41] J. Mann. Wind field simulation. *Probabilistic engineering mechanics*, 13(4):269–282, 1998.

[42] V. Mataix, P. Bucher, R. Zorrilla, R. Rossi, J. Cotela, J. M. Carbonell, M. A. Celigueta, T. Teschemacher, A. Cornejo, C. Roig, G. Casas, M. Masó, S. Warnakulasuriya, M. Núñez, P. Dadvand, S. Latorre, I. Pouplana, J. Irazábal, F. Arrufat, R. Tosi, A. Ghantasala, P. Wilson, D. Baumgaertner, B. Chandra, A. Franci, A. Geiser, K. Bernd, I. Lopez, and J. Gárate. KratosMultiphysics/Kratos: KratosMultiphysics 9.0, 2021.

[43] V. Mataix Ferrándiz. *Innovative mathematical and numerical models for studying the deformation of shells during industrial forming processes with the Finite Element Method*. PhD thesis, Universitat Politècnica de Catalunya, 2020.

[44] V. Mataix Ferrándiz, R. Tosi, I. de Pouplana, R. Zorrilla, L. Gracia, S. Warnakulasuriya, M. Núñez, B. Chandra, A. Ghantasala, P. Bucher, J. Cotela, A. Franci, F. Arrufat, K. B. Sautter, S. Latorre, J. González-Usúa, A. Geiser, P. Dadvand, M. Maso, D. Baumgaertner, M. A. Celigueta, R. Kikkeri Nagaraja, S. Wenczowski, B. Saridar, C. Roig, M. Zidan, and G. Casas. KratosMultiphysics/Examples: Kratos Examples 9.0, 2021.

[45] S. P. Meyn and R. L. Tweedie. Stability of Markovian Processes III: Foster-Lyapunov Criteria for Continuous-Time Processes. Technical Report 3, 1993.

[46] G. Nastac, J. W. Labahn, L. Magri, and M. Ihme. Lyapunov exponent as a metric for assessing the dynamic content and predictability of large-eddy simulations. *Physical Review Fluids*, 2(9), 2017.

[47] F. T. Nieuwstadt, J. Westerweel, and B. J. Boersma. *Turbulence: Introduction to theory and applications of turbulent flows*. Springer, 2016.

[48] N. V. Nikitin. Disturbance growth rate in turbulent wall flows. *Fluid Dynamics*, 44(5), 2009.

[49] E. D. Obasaju. Measurement of forces and base overturning moments on the CAARC tall building model in a simulated atmospheric boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*, 1992.

[50] S. B. Pope. *Turbulent Flows*. Cambridge University Press, aug 2000.

[51] J. Principe, R. Codina, and F. Henke. The dissipative structure of variational multiscale methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 199(13–16):791–801, feb 2010.

[52] P. Sagaut. *Large Eddy Simulations for Incompressible Flows*, volume 3. Springer Berlin, 2000.

[53] E. Tejedor, Y. Becerra, G. Alomar, A. Queralt, R. M. Badia, J. Torres, T. Cortes, and J. Labarta. PyCOMPSs: Parallel computational workflows in Python. *The International Journal of High Performance Computing Applications*, 31(1):66–82, jan 2017.

[54] R. Tosi, R. Amela, R. Badia, and R. Rossi. A parallel dynamic asynchronous framework for Uncertainty Quantification by hierarchical Monte Carlo algorithms. *Journal of Scientific Computing*, 89(28):25, 2021.

[55] T. Von Kármán. Progress in the statistical theory of turbulence. *Proceedings of the National Academy of Sciences of the United States of America*, 34(11):530, 1948.

[56] Q. Wang, S. A. Gomez, P. J. Blonigan, A. L. Gregory, and E. Y. Qian. Towards scalable parallel-in-time turbulent flow simulations. *Physics of Fluids*, 25(11), 2013.