ExaQUte

Exascale Quantification of **U**ncertainties for **Te**chnology and Science Simulation

D5.3. REPORT ON THEORETICAL WORK TO ALLOW THE USE OF MLMC WITH ADAPTIVE MESH REFINEMENT

Document information table

Contract number:	800898
Project acronym:	ExaQUte
Project Coordinator:	CIMNE
Document Responsible Partner:	EPFL
Deliverable Type:	Report
Dissemination Level:	PUblic
Related WP & Task:	WP5 & Task 5.3
Status:	Approved

Authoring

Prepared by CIMNE and EPFL				
Authors	Partner	Modified	Version	Comments
Quentin Ayoul-Guilmard	EPFL	All	1.0.0	Development
Sundar Ganesh	EPFL	All	1.0.0	Development
Marc Nuñez	CIMNE	All	1.0.0	Development
Riccardo Tosi	CIMNE	All	1.0.0	Development
Fabio Nobile	EPFL	All	1.0.0	Supervision
Riccardo Rossi	CIMNE	All	1.0.0	Supervision

Change Log

Versions	Modified Page/Sections	Comments
1.0	All	Submitted version

Approval

Approved by CIMNE and EPFL				
	Name	Partner	Date	OK
Task leader	Fabio Nobile	EPFL	2020-05-30	✓
Coordinator	Riccardo Rossi	CIMNE	2020-05-30	✓

Executive summary

This documents describes several studies undertaken to assess the applicability of Multi-Level Monte Carlo (MLMC) methods to problems of interest; namely in turbulent fluid flow over civil engineering structures. Several numerical experiments are presented wherein the convergence of quantities of interest with mesh parameters are studied at different Reynolds' numbers and geometries.

It was found that MLMC methods could be used successfully for low Reynolds' number flows when combined with appropriate Adaptive Mesh Refinement (AMR) strategies. However, the hypotheses for optimal MLMC performance were found to not be satisfied at higher turbulent Reynolds' numbers despite the use of AMR strategies.

Recommendations are made for future research directions based on these studies. A tentative outline for an MLMC algorithm with adapted meshes is made, as well as recommendations for alternatives to MLMC methods for cases where the underlying assumptions for optimal MLMC performance are not satisfied.

Contents

List of Figures

List of Tables

Nomenclature / Acronym list

Acronym	Meaning
QoI	Quantity of Interest
MLMC	Multi-Level Monte Carlo
MIMC	Multi-Index Monte Carlo
MFMC	Multi-Fidelity Monte Carlo
AMR	Adaptive Mesh Refinement
CVaR	Conditional Value at Risk
MSE	Mean Squared Error
CFL	Courant-Friedrichs-Lewy
PDE	Partial Differential Equation

1 Introduction

1.1 Multi-Level Monte Carlo Methods

MLMC estimators are used in this work for the accurate and efficient estimation of statistics of various QoI. MLMC estimators have shown significant performance improvements over standard Monte Carlo algorithms [11, 15, 17, 20] when tuned properly. There exist different algorithms to calibrate MLMC estimators for simple expectations [8, 16]. MLMC estimators for higher order central moments have also been designed [25]. However, MLMC estimators for parametric expectations like the CVaR and their sensitivities is still an active research area [20]. Progress has been made within the ExaQUte project in work package 6 on tackling the issue of the calculation of CVaR sensitivites. In this report, we focus on the simplest case of computing the first moment or expected value of the QoI.

The construction and main complexity results of MLMC estimators are reviewed hereafter. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space. Let $u(\omega)$ solve a PDE for $\mathbb{P}$-a.e. $\omega \in \Omega$ and $Q \equiv Q(u)$ be the output QoI of which certain statistics are sought, e.g. its expected value.

It is assumed that the underlying PDE is solved using a numerical method with a characteristic discretization parameter h and hence, that one can compute only an approximation Q_h to Q. The standard Monte Carlo estimator for $\mathbb{E}[Q]$ is defined as

$$m^{mc} := \frac{1}{N} \sum_{i=1}^{N} Q_h(\omega^{(i)}), \tag{1}$$

where $\{\omega^{(i)}\}_{i=1}^{N} \in \Omega$ correspond to N i.i.d realizations of the input uncertainties. The accuracy of this estimator can be quantified by looking at the MSE, defined as

$$\text{MSE}(m^{mc}) := \mathbb{E}\left[(m^{mc} - \mathbb{E}[Q])^2\right] = (\mathbb{E}[Q_h - Q])^2 + \frac{\mathbb{V}\text{ar}(Q_h)}{N}. \tag{2}$$

The first term describes the bias error in approximating the QoI and the second term describes the statistical error related to finite sampling. Both error contributions need to be balanced to obtain a good estimate.

It is assumed that there exist positive constants $C_\alpha, \alpha, C_\gamma, \gamma$ such that

$$Cost(Q_h(\omega^{(i)})) \leq C_\gamma h^{-\gamma}, \tag{3a}$$

$$|\mathbb{E}[Q_h - Q]| \leq C_\alpha h^\alpha. \tag{3b}$$

For the MSE to satisfy a tolerance ϵ^2 split equally between the bias and statistical error contributions, it can be shown that the cost to carry out a simulation scales as

$$Cost(m^{mc}) \lesssim \epsilon^{-2-\gamma/\alpha}. \tag{4}$$

MLMC methods aim to improve this complexity by estimating the QoI on a sequence of $L + 1$ meshes of characteristic sizes $h_0 > h_1 > ... > h_L$ such that $h_l = \delta^{-l} h_0$ where $\delta > 1$. It is of interest to estimate the expected value of the QoI on the finest discretization, namely $\mathbb{E}[Q_{h_L}]$. The linearity of the expectations can be used to rewrite it as

$$\mathbb{E}[Q_{h_L}] = \mathbb{E}[Q_{h_0}] + \sum_{l=1}^{L} \mathbb{E}\left[Q_{h_l} - Q_{h_{l-1}}\right] \tag{5}$$

Each of the expectations on the right hand side of Eq. (5) can be independently estimated with a different number of i.i.d samples $\{N_l\}_{l=0}^{L}$, giving rise to the following MLMC estimator.

$$m^{mlmc} = \frac{1}{N_0} \sum_{i=1}^{N_0} Q_{h_0}(\omega^{(i,0)}) + \sum_{l=1}^{L} \frac{1}{N_l} \sum_{i=1}^{N_l} \left[Q_{h_l}(\omega^{(i,l)}) - Q_{h_{l-1}}(\omega^{(i,l)}) \right] \tag{6}$$

$$= \sum_{l=0}^{L} \frac{1}{N_l} \sum_{i=1}^{N_l} Y_l^{(i,l)}, \tag{7}$$

where $Y_0^{(i,0)} := Q_{h_0}(\omega^{(i,0)})$ and $Y_l^{(i,l)} := Q_{h_l}(\omega^{(i,l)}) - Q_{h_{l-1}}(\omega^{(i,l)})$. The MSE of the estimator is defined in a similar fashion to the Monte Carlo estimator.

$$\text{MSE}(m^{mlmc}) := \mathbb{E}\left[(m^{mlmc} - \mathbb{E}[Q])^2 \right] = (\mathbb{E}[Q_h - Q])^2 + \sum_{l=0}^{L} \frac{\mathbb{V}\text{ar}(Y_l)}{N_l}, \tag{8}$$

leading to similar bias and statistical error contribution terms. Assumptions are again made that there exist positive constants $C_\alpha, \alpha, C_\beta, \beta, C_\gamma, \gamma$ such that the following hold.

$$B_l := |\mathbb{E}[Q_{h_l} - Q]| \leq C_\alpha e^{-l\alpha}, \tag{9a}$$

$$V_l := \mathbb{V}\text{ar}(Y_l) \leq C_\beta e^{-l\beta}, \tag{9b}$$

$$C_l := Cost(Y_l^{(i)}) \leq C_\gamma e^{l\gamma}. \tag{9c}$$

The MSE is required to satisfy a tolerance of ϵ^2 that is equally split between bias and statistical error. It can be shown [15] that under these assumptions, the cost of the MLMC simulation goes as

$$Cost(m^{mlmc}) = \sum_{l=0}^{L} N_l C_l \lesssim \begin{cases} \epsilon^{-2} & \beta > \gamma, \\ \epsilon^{-2}(\log \epsilon)^2, & \beta = \gamma, \\ \epsilon^{-2-(\gamma-\beta)/\alpha}, & \beta < \gamma. \end{cases} \tag{10}$$

when $L = L(\epsilon)$ and $N_l = N_l(\epsilon)$ are optimally chosen. For $\beta > \gamma$, the cost is dominated by Monte Carlo sampling on the coarsest levels. For $\beta = \gamma$ the number of samples is distributed evenly across levels and for $\beta < \gamma$, the cost is primarily on the finest levels. Even in the worst case, MLMC estimators are an improvement over standard Monte Carlo in terms of complexity as can be seen when comparing with (4). The assumptions of Eq. (9) will be revisited in Section 1.2 since they have to be properly modified when AMR is used in combination with MLMC.

During implementation, the number of samples N_l and levels L required to attain a given tolerance with optimal complexity depend strongly on the rates α, β and γ. For some problems, these rates are available from theoretical considerations. However in most applications, they are obtained by fitting on estimates of B_l, V_l and C_l respectively. These estimates can be obtained by running a "screening" phase with a few samples or can be estimated and improved on the fly in a continuous manner. The reader is referred to [16, 24] for detailed descriptions of different MLMC algorithms. The ExaQUte MLMC package [2] which was created as a part of deliverable 5.2 contains a multitude of MLMC algorithms to calibrate MLMC estimators. The package is written in a modular manner such that novel MLMC algorithms can be easily tested. It is also written to be agnostic of the solver and has been interfaced with the Kratos Multiphysics package [1, 9] of

the ExaQUte consortium partners at the Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE).

Some notable extensions of the notion of levels are mentioned here for completeness - for example, the notion of fidelity [23] that leads to MFMC algorithms and the notion of continuous levels [11]. MIMC algorithms are also another notable extension of levels to the case of multiple discretization parameters [17].

1.2 Adaptive Mesh Refinement with MLMC

Within the ExaQUte project, it is desired to use AMR techniques and non-uniform meshes to resolve the necessary scales of the problem. The use of AMR techniques poses several challenges for use with MLMC algorithms that have been tackled within this report.

The first is that current MLMC algorithms rely on the identification of a level with a fixed mesh that is used for all samples generated on that level. In most applications with fixed and appropriately chosen sequences of meshes, the hypotheses of Eqs. (9) are satisfied. However, depending on how it is combined with MLMC, AMR can lead to different meshes for different realizations. As a result, the level no longer identifies a fixed mesh. One possibility is to instead identify a level with a given error estimate value. In this case, AMR produces a comparable error estimate for all samples generated at a given level. The corresponding cost, on the other hand, can be stochastic and vary between different realizations. Alternatively, one could identify a level with a given number of degrees of freedom. In this case, the AMR produces meshes with a comparable number of degrees of freedom between all samples generated at a given level, resulting in a comparable cost but stochastic estimated errors. In this report, the former is explored in detail. Future studies are planned to explore the latter strategy as well.

The second challenge is the choice of whether to utilize adapted meshes for each individual sample (stochastic AMR) or whether to use one mesh at each level, adapted with respect to all samples produced at that level (deterministic AMR). Although the latter is computationally less expensive, the challenge is to devise a novel MLMC algorithm that incorporates the new information that additional samples provide to improve the quality of the adapted mesh on the fly. Based on the numerical studies and results of the report, some tentative candidates are outlined in Section 2.3. Both options can be successfully combined with MLMC to yield fully adaptive algorithms. MLMC simulations without hierarchy adaptivity have been implemented successfully in the MLMC Python package [2] of the ExaQUte consortium for use with both options.

We mention that some results exist in literature for combining spatio-temporal adaptivity with MLMC. In [13], an adaptive strategy for MLMC is proposed where local refinement indicators for each sample are averaged to obtain an overall local refinement indicator, which is then used to refine the mesh for the next finer level. Theoretical complexity results are presented in [19] for an adaptive MLMC algorithm based on deterministic AMR. In [18], time step adaptivity for stochastic differential equations is considered in combination with MLMC algorithms. Both deterministic and stochastic AMR are analysed.

1.3 Structure of Report

Two main sets of simulations were run to test the viability of MLMC for use with transition or turbulent flows - one with a lower transitional Reynolds' number of 100 and another with a higher turbulent Reynolds' number of approximately 10^5. In Section 2, 2D transient flow at a Reynolds' number of 100 is simulated over a cylinder and over a rectangle. The mesh convergence of time-averaged lift and drag coefficients are studied. In Section 2.3, these results are extensively discussed and candidates are proposed for algorithms combining MLMC estimators with AMR strategies.

Section 3 explores mesh convergence at a turbulent Reynolds' numbers. Suggestions are made for accelerating simulations using correlated realizations for the case where mesh convergence cannot be observed and, consequently, MLMC cannot be used.

2 Low Reynolds' Number Simulations

2.1 Flow Over Cylinder

2.1.1 Case Description

The following experiments are conducted to assess the convergence of time-averaged QoI with mesh size and degrees of freedom. It is aimed to study whether convergence can be expected for the same Monte Carlo sample across different levels. The flow over a cylinder placed asymmetrically in a channel is studied at a Reynolds' number of 100. In particular, the convergence of time-averaged functionals such as the lift and drag coefficients is studied.

The geometry is a channel without a circular cylinder and can be defined as $\Omega = [0, 2.2] \times [0, 0.41] \backslash B_r(0.2, 0.2), r = 0.05$. The domain can be found in Fig. 1. The flow around the cylinder is governed by the incompressible Navier-Stokes equations -

$$\frac{\partial u}{\partial t} - \nu \Delta u + (u \cdot \nabla)u + \nabla p = 0, \tag{11}$$

$$\nabla \cdot u = 0, \tag{12}$$

where $\nu = 0.001$. The boundary conditions are as follows. The inlet profile is prescribed as

$$u(0, y) = \left(\frac{4Uy(0.41 - y)}{0.41^2}, 0 \right), \tag{13}$$

where U is the peak velocity of the parabolic profile. On the top and bottom walls, no-slip boundary conditions are prescribed. On the outlet, a zero-stress boundary condition is prescribed with the form

$$(\nu \nabla u - pI)n = 0, \tag{14}$$

where n denotes the boundary normal vector and I denotes the identity matrix.

With $U = 1.5$, the mean inlet velocity is $U_{in} = 1.0$. Taking the reference length to be the diameter of the cylinder, the Reynolds' number is

$$\text{Re} = \frac{U_{in}L}{\nu} = \frac{1.0 \times 0.1}{0.001} = 100. \tag{15}$$
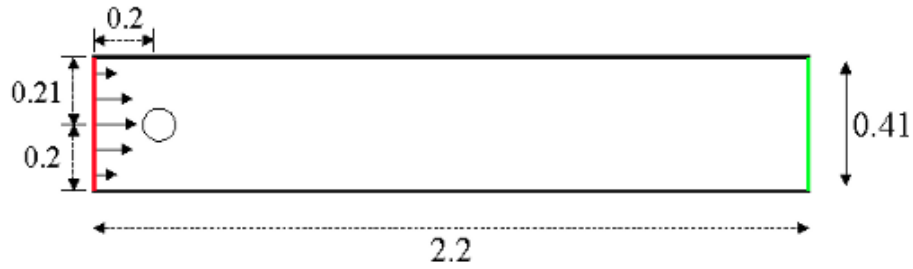
Figure 1: Domain of cylinder problem

Time-averages of the drag coefficients C_D and the lift coefficient C_L are of interest. The lift and drag forces are computed as follows -

$$(F_D, F_L) = \int_S (\nu \nabla u - pI) n \, ds. \tag{16}$$

S denotes the boundary over which the stress is integrated. The coefficients are computed from the forces as

$$C_{D/L} = \frac{F_{D/L}}{\frac{1}{2} U_{in}^2 L}. \tag{17}$$

The time average of these quantities over a window $[T_i, T_f]$ is given by

$$\langle C_{D/L} \rangle = \frac{1}{(T_f - T_i)} \int_{T_i}^{T_f} C_{D/L}(t) dt. \tag{18}$$

In the following sections, the mesh-convergence of these quantities over a sequence of uniform meshes is explored.

2.1.2   Numerical Method

The domain is discretized with triangular finite elements of similar size throughout the domain. Taylor-Hood elements are used for the velocity and pressure fields. The coarsest mesh of those simulated is shown in Fig. 2.

A fractional step method is used to solve the Navier-Stokes equations. In particular, the Chorin-Temam projection method with incremental pressure correction is used. The convective term is treated explicitly. The method involves three steps. In the first step, the momentum equations are solved for an intermediate velocity. In the second step, a Poisson problem is solved for the pressure. In the third step, a final velocity is computed from the intermediate velocity and the pressure field. The complete details of the scheme can be found in [12, Chapter 6]. The resulting linear systems from the first two steps are solved using the stabilized bi-conjugate gradient method and the final projection step using conjugate gradient.

2.1.3   Results

Simulations were carried out for a set of different uniform meshes and time step sizes. The details of the tested meshes are shown in Table 1. The mesh size and time step size
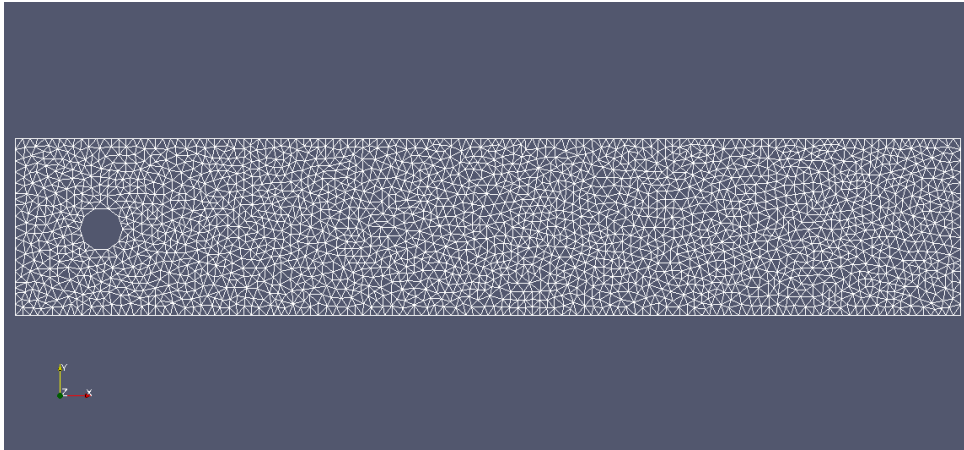
Figure 2: Mesh for cylinder problem

were selected so as to retain the same CFL number defined as

$$\text{CFL} = \frac{U_{in}\Delta t}{h_{min}}, \tag{19}$$

where Δt is the time step size and h_{min} is the minimum mesh size.

No.	DOFs	Elements	Δt	h_{min}
1	21400	4585	1×10^{-3}	1.59×10^{-2}
2	41276	8978	7.07×10^{-4}	1.15×10^{-2}
3	83142	18136	5.0×10^{-4}	7.98×10^{-3}
4	164948	36259	3.54×10^{-4}	5.59×10^{-3}
5	329568	72559	2.5×10^{-4}	3.98×10^{-3}
6	517755	114289	1.77×10^{-4}	3.16×10^{-3}
7	1166040	290288	1.25×10^{-4}	1.99×10^{-3}

Table 1: Mesh parameters tested

The time signals of the lift and drag coefficients computed on Mesh 7 are shown in Fig. 3. The results are compared to a benchmark solution provided along with the the FeatFlow package [3]. As can be seen qualitatively from the diagrams, the mean values as well as the period of oscillation are computed accurately. Based on the time signals, the time window for averaging was chosen from 4.0 to 10.0 seconds. The time-average is computed as the mean of the time signal over this interval.

To study mesh convergence properties, the solution from Mesh 7 is treated as the reference solution and the variation of the quantity $\langle C_{D/L,l} \rangle - \langle C_{D/L,ref} \rangle$, $l \in \{1, 2, ..., 6\}$ with respect to mesh size and degrees of freedom is studied. The resultant plots are shown in Fig. 4 and Fig. 5.

As can be seen from the plots, a geometric decay in the mesh size is observed. The observed convergence rates are shown in Table 2 where a least squares fit was used to fit the data. In view of the convergence rates observed in Table 2, it can be concluded that this simulation can be significantly accelerated using MLMC methods and shows favourable mesh convergence properties.
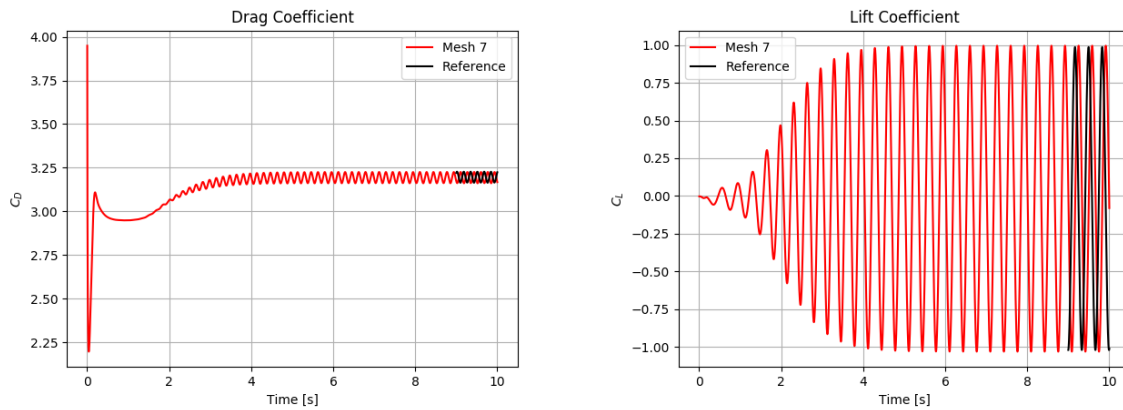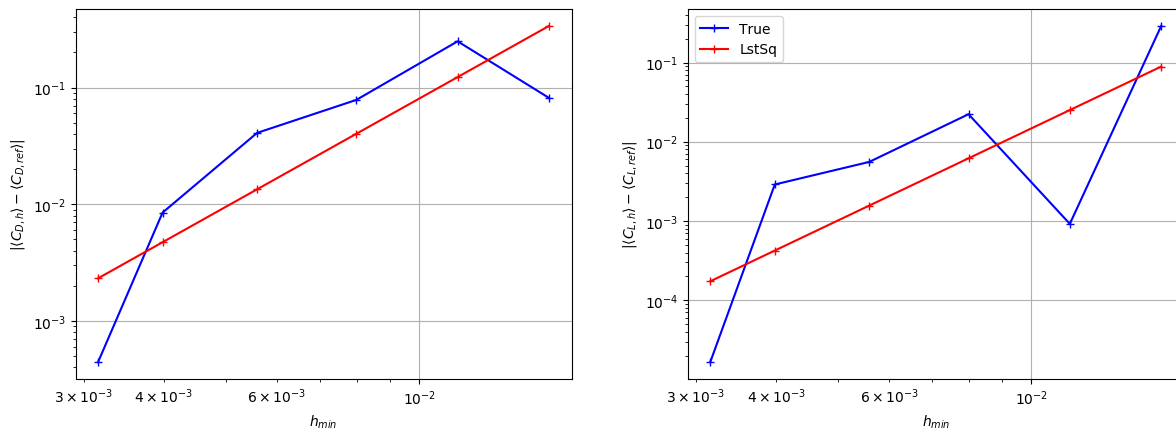
Figure 3: Time signals of lift and drag

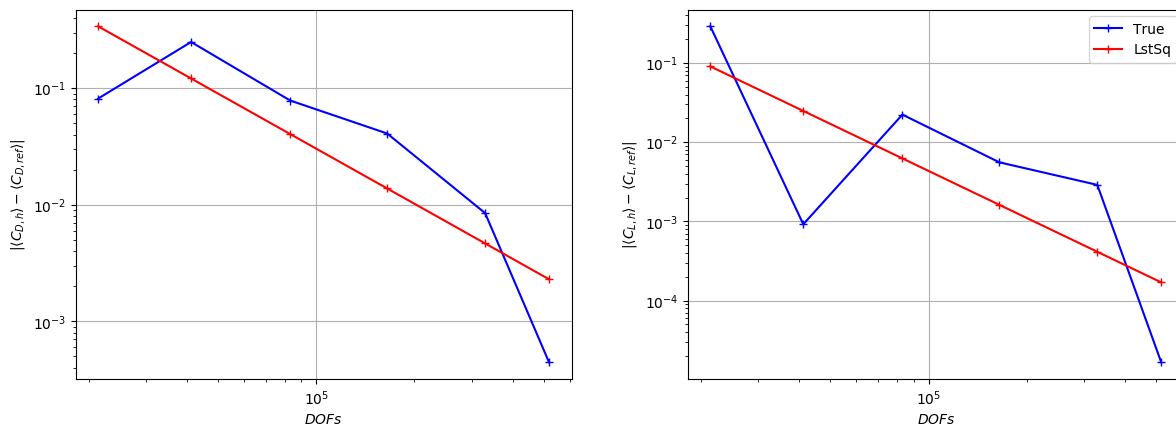Figure 4: Convergence with respect to h_{min}

Figure 5: Convergence with respect to DOFs

2.2 Flow over Rectangle

2.2.1 Case Description

In order to start the development of different solvers and algorithms for the ExaQUte project, a set of benchmarks were defined at the beginning of the project. These can

Results	Least Squares Rate
C_D vs. h_{min}	3.08
C_L vs. h_{min}	3.86
C_D vs. DOFs	-1.57
C_L vs. DOFs	-1.97

Table 2: Convergence rates

be seen in Fig. 6. The eventual goal is to simulate fully turbulent 3D flow over a civil
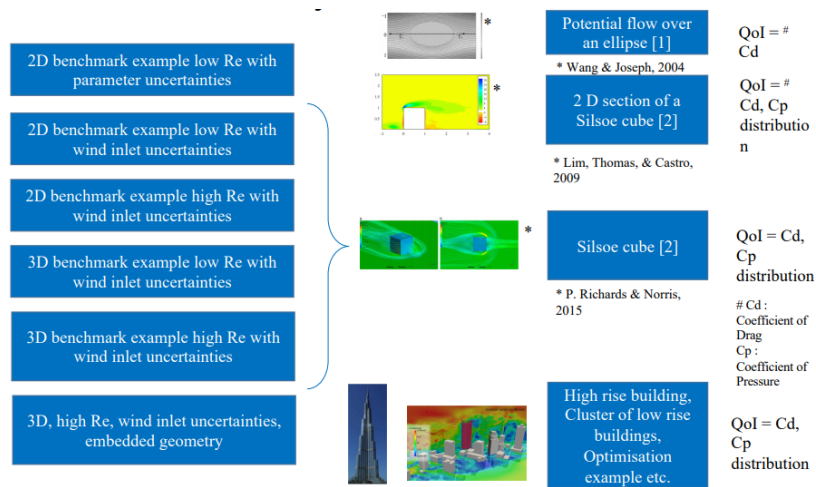

Figure 6: Hierarchy of test cases

engineering structure such as a building.

As an intermediate 2D step, the two-dimensional horizontal cut of the CAARC standard tall building was considered [26]. The dimensions of this example are shown in Fig. 7. The dimensions of the outer domain is 1800×1080. The dimensions of the inner structure is 30×45. However, the cost of computing this example was too high considering the dimensions of the problem and the available hardware constraints. For this reason, a
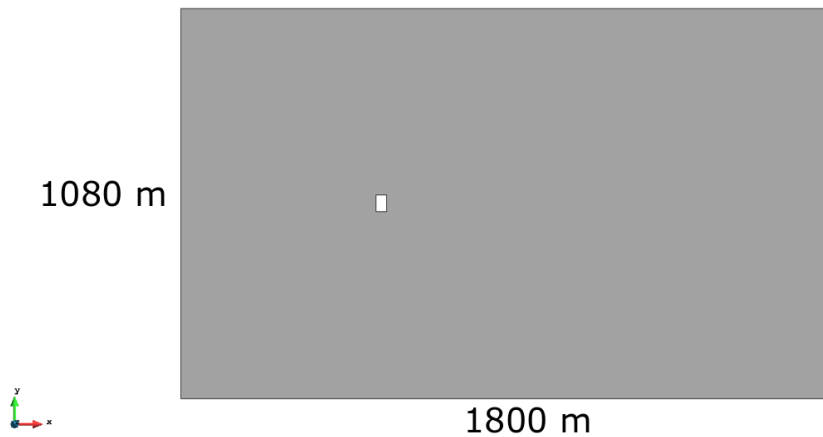

Figure 7: Problem CAARC dimensions. Inner structure 30×45.

reduced but equivalent problem is considered - flow over a 5×1 rectangle. A scheme of
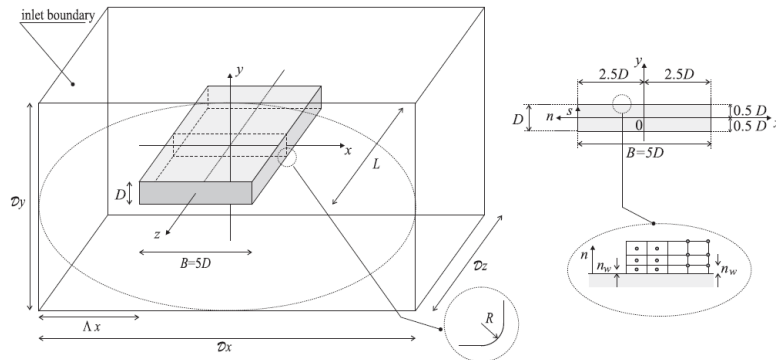
the problem is shown in Fig. 8 and Fig. 9.

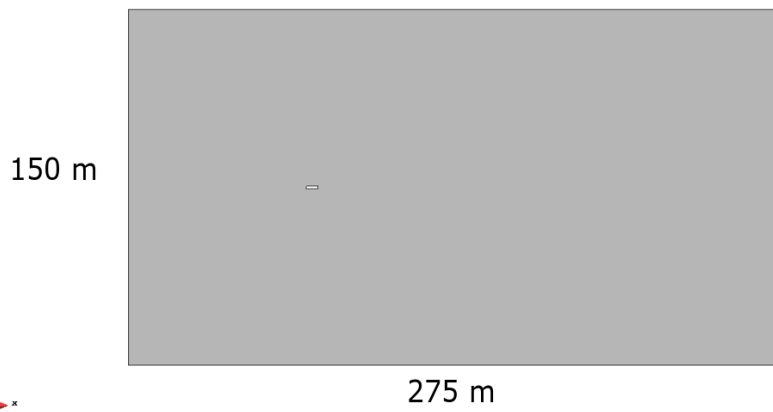Figure 8: Problem description [4], $D = 1$

Figure 9: Rectangle problem dimensions. Inner rectangle 5×1.

The incompressible Navier-Stokes Eqs. 11 and 12 are used to model the fluid flow. A Dirichlet condition is applied at the inlet boundary, where the velocity is prescribed in the horizontal direction and constant along the edge. The inlet velocity is taken to be $v_{inlet} = 2.0$ for all cases considered. This leads to a flow-through time of approximately 140 based on the length of the domain. For time-averaging, the first flow-through time is considered as burn-in time and the corresponding data discarded. The viscosity value is adjusted to achieve the required Reynolds' number in each case. On the upper and lower edges, the outward normal component of velocity is set to zero. On the outlet, a zero stress condition is enforced. No slip boundary conditions are enforced on the surface of the rectangle. The problem is discretized using linear triangular elements for both pressure and velocity fields. Algebraic subgrid scale stabilization is used to stabilize the problem [5, 6]. A second order fractional step method is used for time stepping that treats both pressure and velocity implicitly.

2.2.2 Adaptive Refinement and Mesh Generation

The remeshing tool Mmg [10] provided by the ExaQUte consortium partners at Institut national de recherche en informatique et en automatique (INRIA) is used as the remesh-

ing tool, employing metric-based adaptive refinement. To this end, a nodal metric is constructed in the background mesh that sets the sizes and directions of the new mesh. The metric constructed is of the form

$$\mathcal{M} = \mathcal{R}\hat{\Lambda}^t\mathcal{R} \quad \text{where} \quad \hat{\Lambda} = diag(\hat{\lambda}_i), \tag{20}$$

$$\hat{\lambda}_i = min\left(max\left(\frac{c_d|\lambda_i|}{\varepsilon}, h_{max}^{-2}\right), h_{min}^{-2}\right) \tag{21}$$

and λ_i are the eigenvalues of the Hessian $\mathcal{H}_u$ of a given variable u. The metric depends on the constant c_d and the interpolation error ε. This error is defined as the error that is committed on u by discretizing the domain to obtain a solution u^h on a mesh of size h.

$$\varepsilon = \|u - u^h\| \tag{22}$$

Typically, u is not known and hence, the refinement is performed by prescribing the interpolation error that is committed. It can be shown that this interpolation error is bounded by an expression depending on the hessian of u [14]. In the case of tetrahedrons (denoted by K) with 6 edges, this reads

$$||u - u^h||_{\mathcal{L}_\infty} \le c_d \max_{x \in K} \max_{j=1\ldots6} \langle e_j, |\mathcal{H}_u(x)|e_j \rangle, \tag{23}$$

where the expression $\langle e_j, |\mathcal{H}_u|e_j \rangle$ represents the square of the norm of a vector e in a metric space defined by the hessian of u. In other words, the interpolation error is bounded by the square length of the largest edge of the tetrahedron computed with respect to the metric of the maximal absolute value of the Hessian. This relation allows the inverse operation in adaptive refinement - an interpolation error is prescribed and the optimal edge lengths of the element are found.

The Hessian is estimated numerically by using the shape function gradients of the elements and by performing a volume-based nodal projection twice. For a given scalar quantity, the gradient of the shape functions can be used to compute the gradient on the Gauss points of the scalar quantity. The elemental gradient obtained is projected on the nodes by computing each of its elemental contributions and it is ponderated using the nodal area that is given by the shape functions. Performing the same operation again using the computed nodal gradient instead of the scalar quantity will yield the final estimated Hessian.

The final metric is truncated to minimal and maximal prescribed elemental sizes, h_{min} and h_{max}. This allows the user to control the maximal and minimal sizes that will be generated in the new mesh. For the current case, where an unsteady two-dimensional fluid problem is solved, multiple different choices for u are tested to explore the effect of the AMR strategy. The refinement has been performed by starting with a background mesh generated by GiD [7] with 30000 nodes and a minimal size $h_{min} = 0.01$. The resultant mesh can be seen in Fig. 10. The average velocity field is computed by solving the problem once for a time window of $T = 300$. A snapshot of the averaged velocity field in this mesh and its corresponding computed metric can be seen in Fig. 11 and Fig. 12. The final mesh obtained is shown in Fig. 13 with a closer view of the rectangle in the same mesh in Fig. 14. The main parameters that drive the refinement are selected at the metric construction. These are the minimal size, the maximal size and the interpolation error.
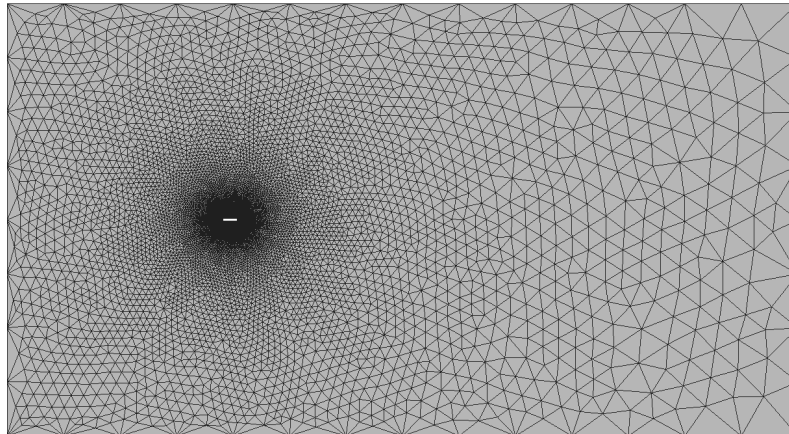
Figure 10: Background mesh generated by GiD. 30000 nodes. $h_{min} = 0.01$
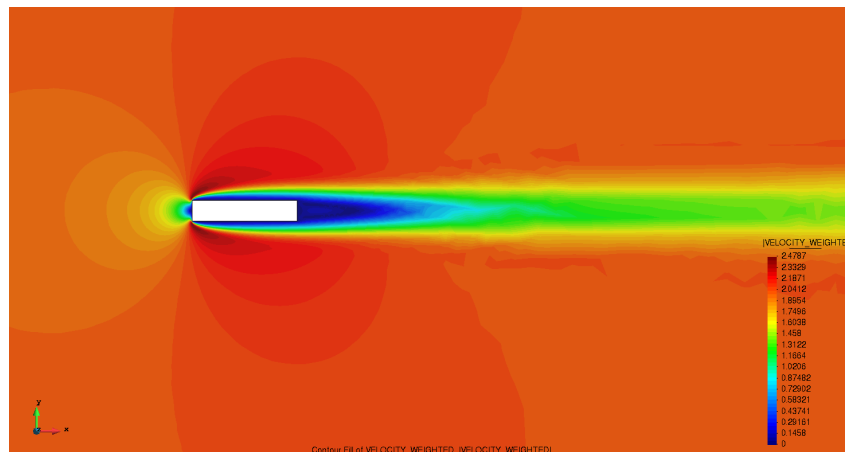
Figure 11: Magnitude of time-averaged velocity field

Figure 12: Norm of nodal metric based on time-averaged velocity field.

- **Minimal and maximal size.** As discussed above, in this approach the minimal and maximal sizes are truncating values, and do not play any role on the computation of the Hessian. If the minimal or maximal size that is prescribed is very far from what the Hessian prescribes, one might observe unphysical refinement. For

Figure 13: Adaptively refined mesh
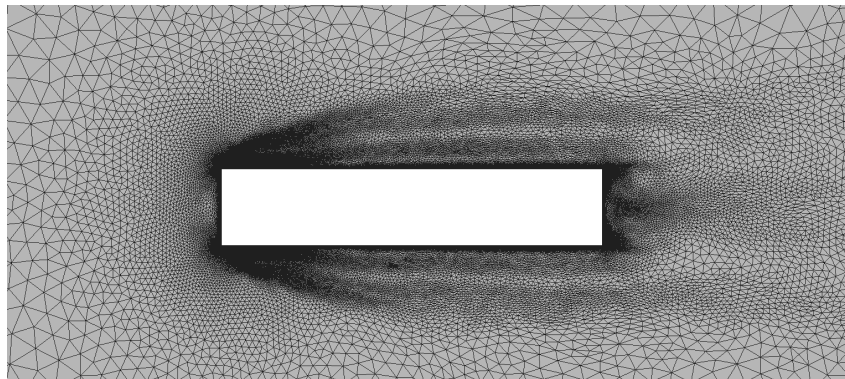
Figure 14: Adaptively refined mesh near rectangle

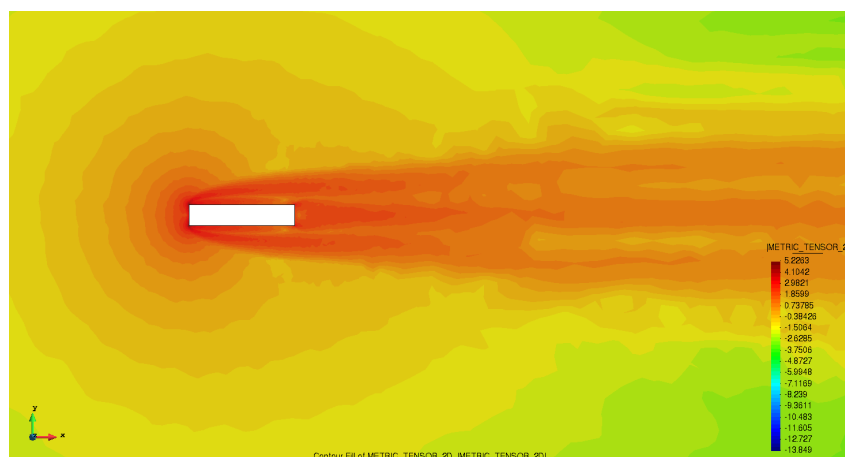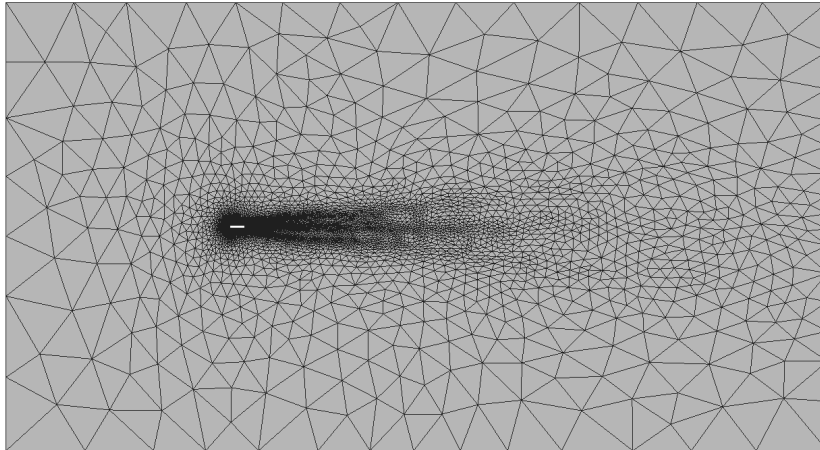this reason, the minimal and maximal size are set at 10^{-12} and 10^{12} respectively, so that the mesh adaptation is performed purely based on the Hessian, and thus purely based on the physics.

- **Interpolation error.** The adaptively refined meshes shown in this section are purely driven by the interpolation error. Thus, the edge length and the number of nodes generated by the remesher depend on the value of ε, as well as on the Hessian, as seen in Eq. (21).

2.2.3 Results

To assess the effects of AMR strategy, a hierarchy of meshes was created by using six different refinement approaches. In order to do so, the same mesh reported in Fig. 10 was used to solve for a flow with $Re = 100$. The velocity and pressure fields computed using this mesh were used to generate the hierarchy of meshes according to six different approaches; namely, mesh refinement was based on a metric computed from -

1. Intersecting the Hessian-based metrics of each time-averaged component of the velocity field

2. Hessian of the time-averaged pressure field

3. Intersecting the Hessian-based metrics of each of the time-averaged velocity components and time-averaged pressure field

4. Intersecting the Hessian-based metrics of the velocity field at every time step and over components

5. Intersecting the Hessian-based metrics of the pressure field at every time step

6. Intersecting the Hessian-based metrics of the velocity component and pressure fields at every time step

The complete list of meshes and their parameter information can be found in Table 5 located in Appendix A. Simulations are conducted wherein all six approaches are used with different interpolation errors and time step sizes. The time averaged drag force at the mesh with interpolation error l, denoted by $\langle F_{D,l} \rangle$, is computed in each simulation. The time averaging is carried out on the window $[140, 300]$. For MLMC algorithms to be applicable to this class of problems, it is important that the hypotheses of Eqs. 9 are fulfilled, where the decay of B_l, V_l and the increase of C_l can be expressed either in terms of the minimal mesh size h_{min} or the nominal interpolation error. To verify this, the convergence of $|\mathbb{E}[F_{D,l} - F_{D,l-1}]|$ is plotted versus the these quantities corresponding to level l.

The resultant plots are shown in Figs. 19, 20, 21, 22, and 23 located in Appendix B. Based on the features of the resultant meshes, it was seen that it is of utmost importance to resolve the wake of the flow behind the rectangle, as well as the corner singularities in the pressure field. In particular, the variation of the level-wise differences with respect to the number of nodes is showcased in Fig. 15 and Fig. 16 for two different time step sizes.
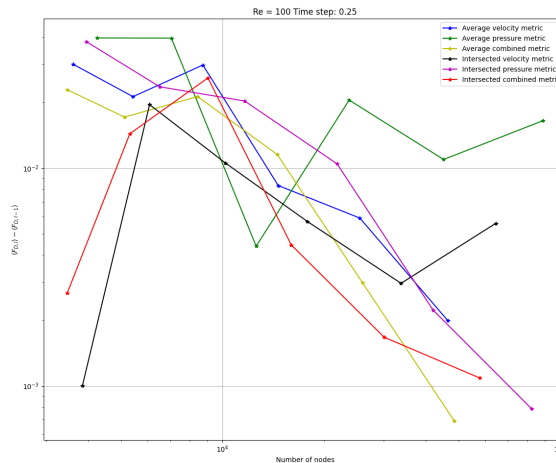
Figure 15: Convergence with respect to number of nodes. $\delta t = 0.25$.

As can be seen from the plots, the choice of AMR strategy strongly affects whether or not the level-wise differences decay, as well as the decay rate. In addition, the best performing strategy was deemed to be that of the intersected pressure field metric. Since this type of decay can be expected for one single sample across different meshes, it is expected that this strategy will also satisfy the MLMC hypotheses of Eqs.(9) expressed in terms of the number of nodes. Given that the cost is also a function of the number of nodes, it can be seen that a good cost rate can be expected for this strategy as well. Additional studies are planned to assess this in further detail.

Figure 16: Convergence with respect to number of nodes. $\delta t = 0.125$.

2.3   Summary and Recommendations for MLMC Algorithms

It was seen from Section 2.1 and 2.2 that under certain circumstances, mesh convergence could be observed for time-averaged QoI when combined with a sequence of uniform meshes or an appropriate AMR strategy. Given that mesh convergence can be obtained fo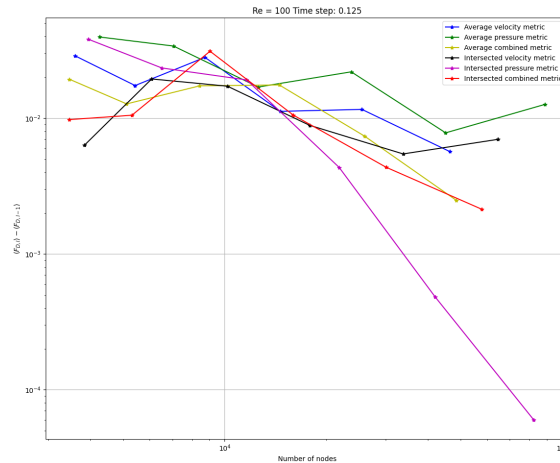r level-wise differences for one given realization, MLMC algorithms can be used for such problems to accelerate simulations. However, the choice still remains as to how to carry out the adaptivity - sample-wise (stochastic) or over all samples (deterministic). In this section, MLMC algorithms are described for both types of adaptivity.

For stochastic AMR, the MLMC algorithm remains similar to the one described in Section 1.1, except for some minor changes. For the ith sample on level l, $Q_l^{(i,l)}$ and $Q_{l-1}^{(i,l)}$ are computed by first solving the underlying PDE on a fixed mesh coarser than level 0 and then successively solving and refining for the same $\omega^{(i,l)}$ until the refinement level l and level $l-1$ respectively. These two samples are then used as correlated realizations in the MLMC estimator for realization $\omega^{(i,l)}$ and the coarser refinements are discarded. In this study, levels are identified with prescribed interpolation errors incurred on the solution. It is planned to explore the identification of levels with prescribed costs or with number of nodes in future studies. Except for this change, the framework of stochastic AMR fits very well within existing MLMC infrastructure and has already been implemented in [2].

For deterministic AMR, the algorithm is more elaborate. Depending on whether screening or continuation type MLMC algorithms are used, different algorithms can be proposed. One form of screening MLMC is to utilize a small number of realizations (~ 20), which are then successively solved on the different levels. Once all of the realizations are solved at a particular level, their metric information can then be intersected to produce a worst-case mesh for the next successive level. This process is carried out until a hierarchy of meshes has been obtained. These samples can as well be used to fit decay rates and constants for the bias, variance and the cost. Such a screening phase is outlined in Algorithm 1. Then, a full MLMC simulation can be run for a hierarchy of sample sites that is optimally selected based on these rates, but with the meshes remaining fixed from the screening phase. The additional samples are then run on these fixed meshes.

However, this excludes the fact that the newly computed samples can be used to successively improve the refined meshes at each level. This naturally leads to a continuation

type strategy as follows. The continuation MLMC strategy relies on computing optimally tuned hierarchies based on a decreasing sequence of tolerances, of which the target tolerance is the final one. This ensures that the algorithm is robust to bad initial estimates of the decay rates and constants. Similarly a continuation strategy can be envisioned for AMR wherein a screening phase is first run to compute an initial hierarchy of meshes. Then, with each successive iteration of the Continuation-MLMC algorithm, the metric information from all new samples at a particular level can be intersected and used to improve the refined mesh at the same level, as well as to provide correlated relations at the corresponding finer level.

Although both options are compatible with the general structure of the MLMC engine [2], implementation and extensive testing of the above algorithms is still required and will be carried out in the future. It is also planned to compare both screening and continuation algorithms to stochastic AMR as well.

Algorithm 1 Deterministic Screening for AMR-MLMC algorithm

Begin with a fixed coarse mesh. Identify this mesh with $l = -1$
Set the number of screening samples N_S
for $l = 0, l \leq L$ **do**
    for $i = 1, i \leq N_S$ **do**
        Compute $Q_{l-1}^{(i,l)}$ on mesh $l - 1$ and store metric information
    end for
    Intersect all metrics and refine to produce level l mesh.
    for $i = 1, i \leq N_S$ **do**
        Compute $Q_l^{(i,l)}$
        Compute $Y_l^{(i,l)} = Q_l^{(i,l)} - Q_{l-1}^{(i,l)}, l > 1 (Y_0^{(i,0)} := Q_0^{(i,0)})$
    end for
end for
Estimate bias, variance and cost decay rates

---

3   High Reynolds' Number Simulations

For the study of higher Reynolds' numbers, the same test case as in Section 2.2 is simulated but with the viscosity adjusted to yield a higher Reynolds' number. The time-averaged velocity field is used to compute metrics for mesh adaptivity. The parameters of the resultant meshes can be seen in Table 3. The simulations are done for $Re = 132719$. In

Interpolation Error	h_{min}	Nodes [$\times 1000$]	CFL	Δt
10^1	0.035	1.1	80	0.7
10^0	0.012	2	80	0.24
10^{-1}	0.0033	5	80	0.066
10^{-2}	0.0011	15	80	0.022
10^{-3}	0.00037	92	80	0.0075

Table 3: Mesh parameters for high Reynolds' number study

contrast to earlier experiments however, the inlet velocity is made stochastic. A total

of 50 realizations of the inlet velocity $v_{inlet} \sim \mathcal{N}(2.0, 0.02)$ are simulated on all of the meshes, which are designed with the 'mean' case of $v_{inlet} = 2.0$. It is of interest to study the convergence of $|\langle F_{D,l}^{(i)} \rangle - \langle F_{D,l-1}^{(i)} \rangle|, i \in \{1, 2, ..., 50\}$ for each of the samples. It is also important to study whether the quantity

$$|\mathbb{E}\left[\langle F_{D,l} \rangle - \langle F_{D,l-1} \rangle\right]| \tag{24}$$

converges in l as well. The expectation is estimated using Monte Carlo sampling as

$$\mathbb{E}\left[\langle F_{D,l} \rangle\right] \approx \frac{1}{N} \sum_{i=1}^{N} \langle F_{D,l}^{(i)} \rangle, \tag{25}$$

where N represents the number of realizations.

Fig. 17 shows $\mathbb{E}[\langle F_{D_l} \rangle - \langle F_{D_{l-1}} \rangle]$ versus the interpolation error, together with all of the realizations to estimate it. It can be seen from the plot that geometric decay of this quantity in the mesh is not observed. The variation of the variance of the differences
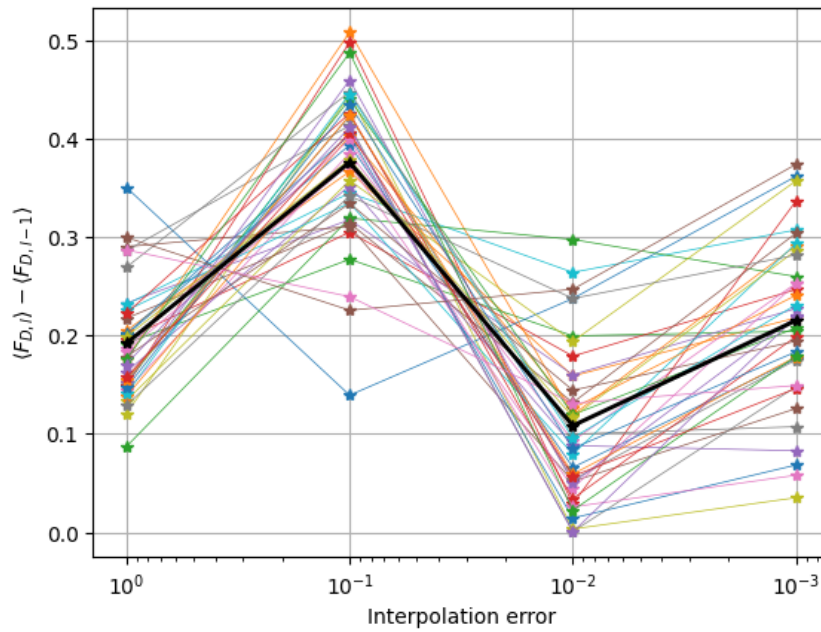

Figure 17: Bias decay plot. Same colour for same realization at different levels. Black line denotes sample average. Levels defined by Table 3.

$\mathbb{V}ar[\langle F_{D_l} \rangle - \langle F_{D_{l-1}} \rangle]$ is also plotted in Fig. 18. As can be seen from the figure, this quantity does not decay with the mesh parameters.

Level	$l = 4$	$l = 3$	$l = 2$	$l = 1$
$l = 4$	1	0.59	0.52	0.65
$l = 3$	-	1	0.32	0.49
$l = 2$	-	-	1	0.36
$l = 1$	-	-	-	1

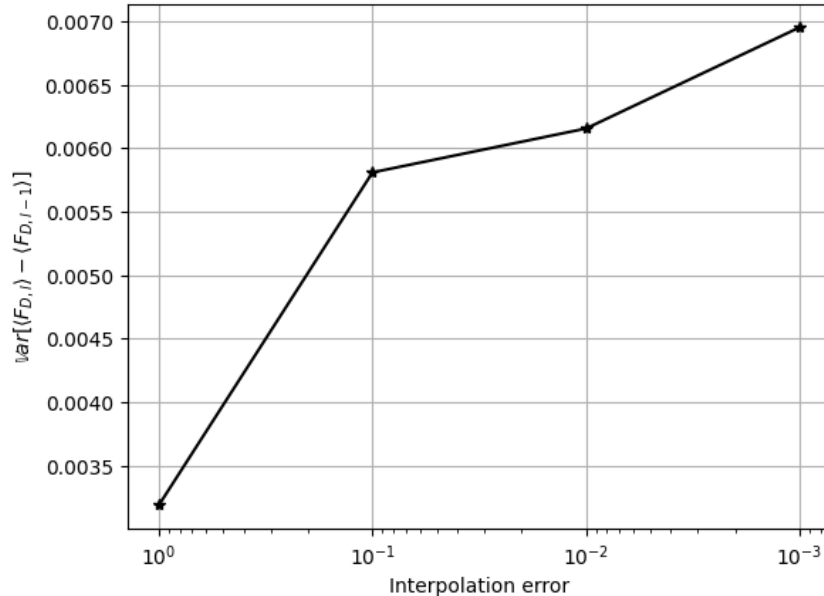Table 4: Correlation matrix across levels

Figure 18: Variance decay plot. Levels defined by Table 3

On further investigation, it was found that bias and variance decay could be difficult to obtain at high Reynolds' numbers due to the chaotic nature of the flow. This makes it very difficult to retain pathwise correlation of both fine and coarse samples. As a potential alternative to MLMC methods, multifidelity and control variate estimators were explored [21, 22]. These methods rely on computing correlated realizations across different 'fidelities'. Possible candidates for fidelity include mesh resolution, low rank approximations, less descriptive models, etc. Specifically, no assumptions of the type in Eqs. (9) are made. To assess their viability in the context of AMR, cross-level correlations were computed using the computed samples. The resultant correlation matrix is shown in Table 4. Based on the results of [21], it was found that a speed-up of approximately 40% could be obtained when using the level 1 mesh as a control variate for the QoI on the level 4 mesh. Additional studies are under way to assess the use of more complex multifidelity estimators.

4   Conclusions and Future Scope

Numerous experiments were conducted to assess the applicability of MLMC algorithms to turbulent or transitional fluid flow problems with AMR. Three cases were tested, namely the following -

1. Flow around a cylinder at $Re = 100$

2. Flow around a rectangle at $Re = 100$

3. Flow around a rectangle at $Re = 1.3 \times 10^5$

To verify the hypotheses required for MLMC to work at optimal complexity, the mesh convergence of time-averaged quantities of interest was studied.

In the first case, it was found that mesh convergence could be observed with uniform refinement with close to optimal rates. It was concluded that this problem stood to benefit from the use of MLMC algorithms. In the second case, an additional challenge was applied in the form of sharp corners in the geometry. It is well known that the Navier-Stokes equations give rise to singularities in the pressure field on sharp corners of the geometry. As a result, this problem proved to be a viable testing ground for adaptive meshing strategies. With an optimally selected AMR strategy, it was concluded that mesh-convergence of time-averaged quantities of interest could be observed and that MLMC method could be used in combination with AMR for problems of this specific class, although not without challenges. In the third case, another challenge was applied in the form of a higher Reynolds' number. Due to the turbulent nature of the flow and the difficulty in resolving corner singularities and the wake adequately, no mesh convergence could be observed. As a result, it was concluded that MLMC simulations could not be applied in this scenario. However, MFMC methods were shown to be a possible alternative candidate for accelerating simulations using correlated realizations. However, it still remains to be seen whether alternative AMR strategies, solution algorithms for the Navier-Stokes equations or geometry characteristics could lead to a different conclusion for simulations at similar high Reynolds' numbers.

Section 2.3 outlines two AMR-MLMC algorithms. In one algorithm, AMR is carried out for each individual sample. In the other, AMR is carried out using the intersected metric information from all samples. It was commented that although the ExaQUte MLMC library [2] was capable of carrying out both types of simulations, that fully adaptive strategies were yet to be defined in detail and tested. Extensive studies are planned to compare these two algorithms and to study their behaviour, as well as to design novel MLMC algorithms based on observed mesh convergence behaviour.

It is also planned to explore more extensively the different combinations of factors that contributed to mesh convergence. Some of these factors are the AMR strategy, the solution algorithm of Navier-Stokes equations and how the functionals in space and time are computed. In addition, an eventual longer-term aim of the project is to use goal-oriented AMR techniques. These techniques aim to utilize adjoint-information to adapt the meshes. Although adjoint-based adaptivity is feasible for lower Reynolds' numbers, it still poses a significant challenge for higher Reynolds' numbers due to the chaotic nature of the problem.

References

[1] Migration of a generic multi-physics framework to HPC environments. *Computers and Fluids*, 80(1):301–309, 2013. ISSN 00457930. doi:10.1016/j.compfluid.2012.02.004.

[2] R. Amela, Q. Ayoul-Guilmard, R. M. Badia, S. Ganesh, F. Nobile, R. Rossi, and R. Tosi. ExaQUte XMC, doi:10.5281/zenodo.3235833, 2019.

[3] E. Bayraktar, O. Mierka, and S. Turek. Benchmark computations of 3d laminar flow around a cylinder with cfx, openfoam and featflow. *International Journal of Computational Science and Engineering*, 7(3):253–266, 2012.

[4] 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, 2014.

[5] R. Codina. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Computer Methods in Applied Mechanics and Engineering*, 156(1-4):185–210, apr 1998. ISSN 00457825. doi:10.1016/S0045-7825(97)00206-5.

[6] R. Codina. Pressure Stability in Fractional Step Finite Element Methods for Incompressible Flows. *Journal of Computational Physics*, 2001. ISSN 00219991. doi:10.1006/jcph.2001.6725.

[7] A. Coll, R. Ribó, M. Pasenau, E. Escolano, J. Perez, A. Melendo, A. Monros, and J. Gárate. Gid v.14 user manual. 2018.

[8] N. Collier, A.-L. Haji-Ali, F. Nobile, E. Von Schwerin, and R. Tempone. A continuation multilevel monte carlo algorithm. *BIT Numerical Mathematics*, 55(2):399–432, 2015.

[9] 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.

[10] 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. ISSN 10902716. doi:10.1016/j.jcp.2014.01.005.

[11] G. Detommaso, T. Dodwell, and R. Scheichl. Continuous level monte carlo and sample-adaptive model hierarchies. *SIAM/ASA Journal on Uncertainty Quantification*, 7(1):93–116, 2019.

[12] J. Donea and A. Huerta. *Finite element methods for flow problems*. John Wiley & Sons, 2003.

[13] M. Eigel, C. Merdon, and J. Neumann. An adaptive multilevel monte carlo method with stochastic bounds for quantities of interest with uncertain data. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1219–1245, 2016.

[14] P. Frey and F. Alauzet. Anisotropic mesh adaptation for cfd computations. *Computer Methods in Applied Mechanics and Engineering*, 194(48):5068 – 5082, 2005. ISSN 0045-7825. doi:https://doi.org/10.1016/j.cma.2004.11.025. URL `http://www.sciencedirect.com/science/article/pii/S0045782505000794`. Unstructured Mesh Generation.

[15] M. B. Giles. Multilevel monte carlo path simulation. *Operations Research*, 56(3): 607–617, 2008.

[16] M. B. Giles. Multilevel monte carlo methods. *Acta Numerica*, 24:259–328, 2015.

[17] A.-L. Haji-Ali, F. Nobile, and R. Tempone. Multi-index monte carlo: when sparsity meets sampling. *Numerische Mathematik*, 132(4):767–806, 2016.

[18] H. Hoel, E. Von Schwerin, A. Szepessy, and R. Tempone. Implementation and analysis of an adaptive multilevel monte carlo algorithm. *Monte Carlo Methods and Applications*, 20(1):1–41, 2014.

[19] R. Kornhuber and E. Youett. Adaptive multilevel monte carlo methods for stochastic variational inequalities. *SIAM Journal on Numerical Analysis*, 56(4):1987–2007, 2018.

[20] S. Krumscheid and F. Nobile. Multilevel monte carlo approximation of functions. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1256–1293, 2018.

[21] L. W. Ng and K. E. Willcox. Multifidelity approaches for optimization under uncertainty. *International Journal for numerical methods in Engineering*, 100(10):746–772, 2014.

[22] B. Peherstorfer, K. Willcox, and M. Gunzburger. Optimal model management for multifidelity monte carlo estimation. *SIAM Journal on Scientific Computing*, 38(5): A3163–A3194, 2016.

[23] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *Siam Review*, 60(3):550–591, 2018.

[24] M. Pisaroni. Multi level monte carlo methods for uncertainty quantification and robust design optimization in aerodynamics. Technical report, EPFL, 2017.

[25] M. Pisaroni, S. Krumscheid, and F. Nobile. Quantifying uncertain system outputs via the multilevel Monte Carlo method — Part I: central moment estimation. Technical Report 23.2017, MATHICSE, 2017.

[26] R. Wardlaw and G. Moss. A standard tall building model for the comparison of simulated natural winds in wind tunnels. *CAARC, CC 662m Tech*, 25, 1970.

A   Mesh Parameters

Table 5: List of meshes used in adaptive refinement study for the flow over a rectangle at $Re = 100$. Error scaled in each approach to accomplish comparable number of nodes and minimal mesh size.

Approach	Interpolation error	Minimal mesh size	Number of nodes
1	0,05	0,005527	2605
1	0,025	0,00333	3606
1	0,0125	0,002778	5427
1	0,00625	0,00181	8742
1	0,003125	0,001225	14619
1	0,001563	0,000831	25510
1	0,000781	0,000434	46504
1	0,000391	0,000418	86954
1	0,000195	0,000264	166481
2	0,015	0,001171	2770
2	0,0075	0,000783	4254
2	0,00375	0,00052	7049
2	0,001875	0,00032	12574
2	0,000937	0,000254	23714
2	0,000469	0,000156	45163
2	0,000234	0,000109	88926
2	0,000117	0,000096	132884
2	0,000059	0,000055	344028
3	0,065	0,002387	2615
3	0,0325	0,001489	3468
3	0,01625	0,001262	5123
3	0,008125	0,00084	8433
3	0,004063	0,000604	14517
3	0,002031	0,000387	25981
3	0,001016	0,000273	48579
3	0,000508	0,000189	92884
3	0,000254	0,000129	181259
4	750	0,005071	2624
4	375	0,003401	3843
4	187,5	0,003151	6076
4	93,75	0,001859	10194
4	46,875	0,001427	17844
4	23,4375	0,000965	33786
4	11,71875	0,000688	64524
4	5,859375	0,000464	127064
4	2,929688	0,000328	246803
5	225	0,001206	2655
5	112,5	0,00093	3947

5	56,25	0,000619	6516
5	28,125	0,000434	11624
5	14,0625	0,000249	21838
5	7,03125	0,000183	42042
5	3,515625	0,000128	82410
5	1,757812	0,000101	162998
5	0,878906	0,000069	322375
6	975	0,002079	2506
6	487,5	0,001474	3466
6	243,75	0,001123	5311
6	121,875	0,000943	9048
6	60,9375	0,000398	15976
6	30,46875	0,000365	30097
6	15,234375	0,000243	57853
6	7,617188	0,000168	113982
6	3,808594	0,000123	222476

B   Mesh Convergence Plots
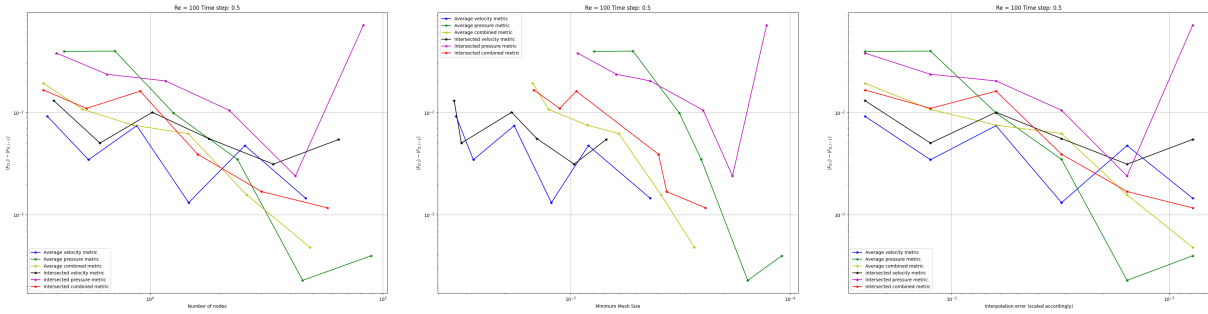
Figure 19: Number of nodes, the minimal size and the interpolation error (left, center and right respectively). $\delta t = 0.5$
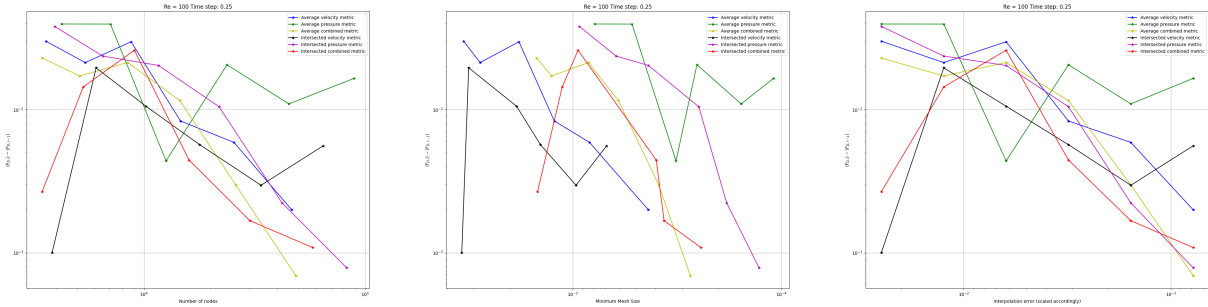

Figure 20: Number of nodes, the minimal size and the interpolation error (left, center and right respectively). $\delta t = 0.25$.
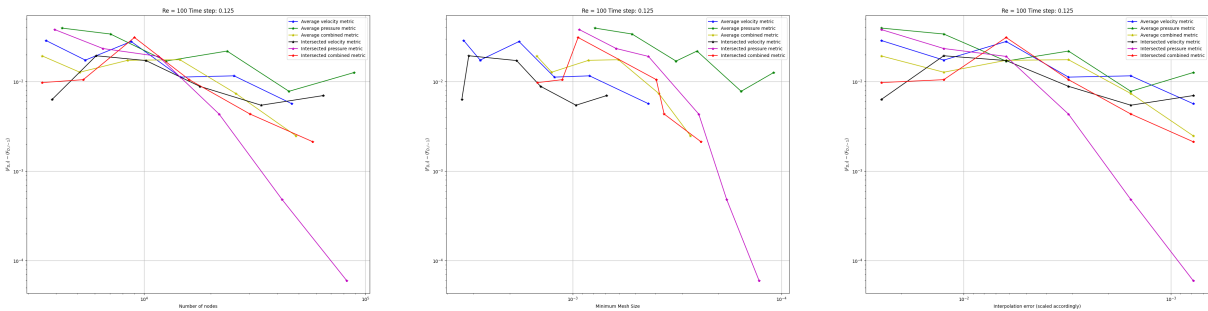

Figure 21: Number of nodes, the minimal size and the interpolation error (left, center and right respectively). $\delta t = 0.125$.
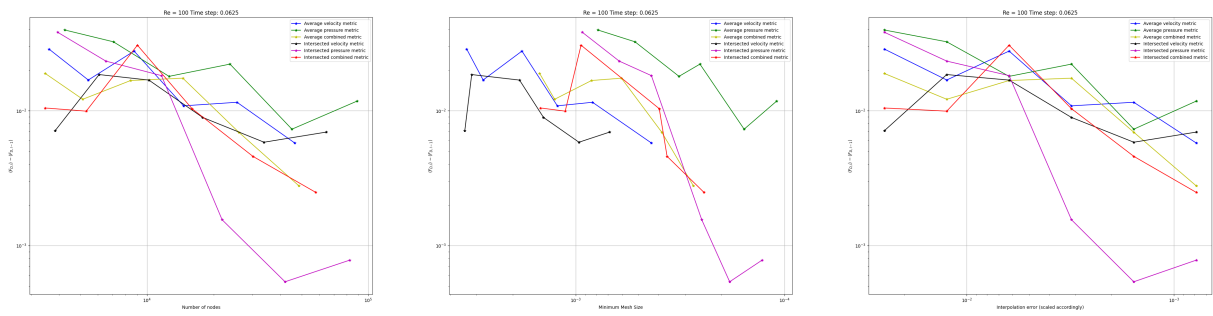
Figure 22: Number of nodes, the minimal size and the interpolation error (left, center and right respectively). $\delta t = 0.0625$.
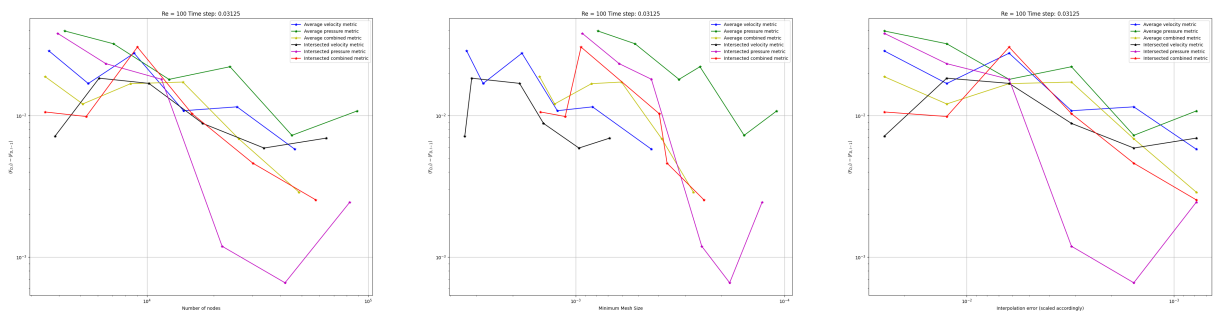

Figure 23: Number of nodes, minimal mesh size and the interpolation error (left, center and right respectively). $\delta t = 0.03125$.