

# Parallel adaptive mesh refinement for large-eddy simulations of turbulent flows

O.Antepará<sup>a,\*</sup>, O.Lehmkuhl<sup>a,b</sup>, R.Borrell<sup>b</sup>, J.Chiva<sup>a</sup>, A.Oliva<sup>a,\*</sup>

<sup>a</sup>*Heat and Mass Transfer Technological Center, ETSEIAT, Technical University of Catalonia, C/ Colom 11, 08222, Terrassa, Spain*

<sup>b</sup>*Termo Fluids, S.L., c/ Magí Colet 8, 08204 Sabadell (Barcelona), Spain*

---

## Abstract

In this paper a parallel adaptive mesh refinement (AMR) strategy for large eddy simulations (LES) of turbulent flows is presented. The underlying discretization of the Navier-Stokes equations is based on a finite-volume symmetry-preserving formulation, with the aim of preserving the symmetry properties of the continuous differential operators and ensure both, stability and conservation of kinetic-energy balance. The conservation properties are tested for the meshes resulting from the AMR process, which typically contain transitions between zones with different level of refinement. Our AMR scheme applies a cell-based refinement technique, with a physics-based refinement criteria based on the variational multi-scale (VMS) decomposition theory. The overall AMR process, from the selection of the cells to be refined/coarsened till the pre-processing of the resulting mesh, has been implemented in a parallel code, for which the parallel performance has been attested on an AMD Opteron based supercomputer. Finally, the robustness

---

\*Corresponding author. Tel.: +34 93 739 81 92; fax:+34 93 739 89 20.  
*Email addresses:* `cttc@cttc.upc.edu` (A.Oliva )

and accuracy of our methodology is shown on the numerical simulation of the turbulent flow around a square cylinder at  $Re = 22000$  and the turbulent flow around two side-by-side square cylinders at  $Re = 21000$ .

*Key words:* Parallel adaptive mesh refinement, Turbulent flow around bluff bodies, LES, physics-based criteria.

---

## 1. Introduction

The direct numerical simulation (DNS) of incompressible turbulent flows is limited by the wide range of scale motions that need to be accurately solved. In each zone of the simulation domain, the mesh needs to be dense enough in order to solve the smallest scales of motion and, at the same time, all these discrete elements become coupled by the largest scales of motion. The result is a large discrete system of mutually coupled variables that, commonly requires unaffordable computing resources in order to solve it.

This situation, has prompted the scientific community to develop strategies in order to reduce the computing requirements. An option is the large eddy simulation (LES), based in modeling the subgrid scales of motion and therefore, allowing to coarse the mesh. Another strategy consists in optimize the mesh generation in order to avoid unnecessary zones of refinement. Adaptive mesh refinement (AMR) methods focus in this second aspect by dynamically refine or coarsen any part of the mesh according to the flow problem. The desired benefit from these techniques is an automatic and dynamic mesh adaptation to accurately solve any flow, minimizing the number of grid cells. Note that, as a result, this methods also cancel the cost of “manually” generating a suitable mesh for the solution of the flow, what is

becoming a tedious problem on the HPC context.

At first, the AMR techniques were initially introduced by Berger [1, 2], and Powell [3] whom described an AMR formulation for Cartesian meshes and cell based AMR methods, respectively. On the context of AMR applied to flow around bluff bodies, finite element AMR approaches have been developed using a posteriori error estimation based on the residuals of the Navier Stokes equations [4]. In [5, 6] a posteriori error estimation for turbulent flow is considered on applications like the flow around a surface mounted cube and a square cylinder. Another AMR approach was developed for engineering problems by Berrone et al. [7], where the viability of a fully combined space and time adaptivity for engineering problems was investigated. Although the large number of numerical studies available, most of them are based on error control technique, applied on a finite element framework. Only few studies have been performed using solution-directed mesh-refinement methods or applied to finite volume framework for different applications [8, 9, 10, 11].

It is also important that solution codes achieve good parallel performance in current supercomputers in order to take advantage of the increasingly available computing power. In this regard, the development of parallel AMR algorithms is mandatory, although important difficulties appear such as the global labeling of the unknowns, the treatment of the elements at the boundaries of the mesh subdomains or the achievement of a good workload distribution.

In this context the aim of the present work has been the development of a parallel AMR method to be applied in LES of turbulent flow at high Reynolds number, using a physics-based refinement criteria in a finite vol-

ume framework with conservative formulations. This has been implemented on the top of the TermoFluids (TF) CFD software platform [12]. In TF the Navier-Stokes equations are discretized following a symmetry-preserving formulation [13, 14, 15], thus, the conservation properties on the meshes resulting from the AMR process, which typically contain transitions between zones with different level of refinement, has been analyzed. The WALE model [16] is used within a variational multiscale framework [17] to deal with the smallest scales of motion. Furthermore, AMR refinement criteria based on the VMS scale separation theory has been developed and the AMR algorithm has been implemented in a parallel code, for which the parallel performance has been attested on an AMD Opteron based supercomputer. The robustness of our method has been proven on the numerical simulation of the flow around square cylinder at Reynolds number 22000 and the flow around two side-by-side square cylinders at  $Re=21000$ . These cases cover the main turbulent flows features such as flow separation, vortex shedding and appearance of vortex in the wake of the cylinder [18, 19, 20, 21, 22, 23, 24].

The rest of the paper is organized as follows. In Section 2, the system of governing equations using a symmetry preserving discretization is described. In Section 3, a detailed description of the adaptive mesh refinement scheme and an ongoing parallelization strategy with a performance study are presented. Moreover, conservation test are carried out to test the AMR mesh on a Rankine vortex problem and a detailed description of the refinement criteria with its corresponding applications on different problems are presented. In Section 4, the solutions for a turbulent flow around a square cylinder and two side-by-side square cylinders are compared to experimental and numeri-

cal results. Finally, some conclusions are drawn.

## 2. Mathematical Formulation

In large-eddy simulations (LES) the spatial filtered and discretized Navier-Stokes equations are defined as

$$\mathcal{M}\bar{\mathbf{u}} = 0 \quad (1)$$

$$\Omega \frac{\partial \bar{\mathbf{u}}}{\partial t} + \mathbf{C}(\bar{\mathbf{u}})\bar{\mathbf{u}} + \nu \mathbf{D}\bar{\mathbf{u}} + \rho^{-1} \Omega \mathbf{G}\bar{\mathbf{p}} = \mathbf{C}(\bar{\mathbf{u}})\bar{\mathbf{u}} - \overline{\mathbf{C}(\mathbf{u})\mathbf{u}} \approx -\mathcal{M}\mathcal{T} \quad (2)$$

where  $\bar{\mathbf{u}}$  and  $\bar{\mathbf{p}}$  represent the filtered velocity vector and pressure, respectively,  $\rho$  is the fluid density and  $\nu$  is the kinematic viscosity,  $\Omega$  is a diagonal matrix with the sizes of control volumes. Convective and diffusive operators in the momentum equation for the velocity field are given by  $\mathbf{C}(\bar{\mathbf{u}}) = (\bar{\mathbf{u}} \cdot \nabla)$  and  $\mathbf{D} = -\nabla^2$ , respectively. Gradient and divergence operators are given by  $\mathbf{G} = \nabla$  and  $\mathcal{M} = \nabla \cdot$ , respectively. The term that requires modelling is the filtered non-linear convective term.  $\mathcal{T}$  is the SGS stress tensor, which is defined as [25],

$$\mathcal{T} = -2\nu_{sgs}\overline{\mathcal{S}}_{ij} + (\mathcal{T} : \mathbf{I})\mathbf{I}/3 \quad (3)$$

$$\overline{\mathcal{S}}_{ij} = \frac{1}{2}[\mathbf{G}(\bar{\mathbf{u}}) + \mathbf{G}^*(\bar{\mathbf{u}})] \quad (4)$$

where  $\overline{\mathcal{S}}_{ij}$  is the rate-of-strain tensor and  $\mathbf{G}^*$  is the transpose of the gradient operator.

To close the formulation, a suitable expression for the subgridscale (SGS) viscosity, must be introduced. LES studies have been performed using a SGS

model suitable for unstructured formulations: the wall-adapting local-eddy viscosity model within a variational multi-scale framework (VMS-WALE) [16, 17]. A brief description of this model is given hereafter.

*2.1. Wall-adapting eddy viscosity model within a variational multiscale framework (VMS-WALE)*

The variational multi-scale (VMS) concepts for Large Eddy Simulation (LES) was originally formulated by Hughes et al. [17] in the Fourier space, and is a viable and practical approach for LES of turbulent flows. In VMS the decomposition of the flow into three scales is considered: large scales, resolved small scales and unresolved small scales. If a explicit filter is introduced, a splitting of the resolved scales can be performed,

$$f' = \bar{f} - \hat{f} \quad (5)$$

where following Vreman [26] notation, the large scales are determined by  $\hat{f}$ , the small scales by  $f'$  and  $\bar{f}$  is the original resolved quantity. Thus, for the large-scale parts of the resolved  $\bar{\mathbf{u}}$ , a general governing equation can be derived,

$$\Omega \frac{\partial \bar{\mathbf{u}}}{\partial t} + \mathbf{C}(\bar{\mathbf{u}}) \bar{\mathbf{u}} + \nu \mathbf{D} \bar{\mathbf{u}} + \rho^{-1} \Omega \mathbf{G} \bar{p} - \bar{f} = -\frac{\partial \hat{\mathcal{T}}}{\partial x_j} - \frac{\partial \mathcal{T}'}{\partial x_j} \quad (6)$$

Here,  $\hat{\mathcal{T}}$  is the subgrid large-scale term and  $\mathcal{T}'$  is the subgrid small-scale term. Now, assuming that the unresolved scales doesn't have any effect on the large scale equation ( $\hat{\mathcal{T}} \approx 0$ ), it is only necessary to model the effect of the small scale term  $\mathcal{T}'$ . In our implementation the small-small strategy is used in conjunction with the wall-adapting eddy viscosity (WALE) model [16]:

$$\begin{aligned}
\mathcal{T}' &= -2\nu_{sgs}\mathcal{S}'_{ij} + \frac{1}{3}\mathcal{T}'\delta_{ij} \\
\nu_{sgs} &= (C_w^{vms}\Delta)^2 \frac{(\mathcal{V}'_{ij} : \mathcal{V}'_{ij})^{\frac{3}{2}}}{(\mathcal{S}'_{ij} : \mathcal{S}'_{ij})^{\frac{5}{2}} + (\mathcal{V}'_{ij} : \mathcal{V}'_{ij})^{\frac{5}{4}}} \\
\mathcal{S}'_{ij} &= \frac{1}{2}[\mathbf{G}(\bar{\mathbf{u}}') + \mathbf{G}^*(\bar{\mathbf{u}}')] \\
\mathcal{V}'_{ij} &= \frac{1}{2}[\mathbf{G}(\bar{\mathbf{u}}')^2 + \mathbf{G}^*(\bar{\mathbf{u}}')^2] - \frac{1}{3}[\mathbf{G}(\bar{\mathbf{u}}')^2\mathbf{I}]
\end{aligned} \tag{7}$$

where  $C_w^{vms}$  is the equivalent of the WALE coefficient for the *small-small* VMS approach and for finite volume method its value lies between 0.3 and 0.5 [27].

## 2.2. Numerical method

Second-order spectro-consistent schemes on a collocated unstructured grid arrangement were adopted for the discretization of the governing equations. It is remarkable that those schemes are conservative, i.e. they preserve the symmetry properties of the continuous differential operators and ensure both, stability and conservation of the kinetic- energy balance even at high Reynolds numbers and with coarse grids [28, 29]. For the discretization in time of the momentum equation a two-step linear explicit scheme on a fractional-step method was used for the convective and diffusive terms [30], while the pressure is solved using an implicit first-order scheme. This methodology has been extensively tested and verified with accurate results for solving the flow over bluff bodies with massive separation [14, 15, 31]. Computations were carried out using meshes generated by a constant step extrusion of a two-dimensional (2D) grid. Hence, the spanwise coupling of

the discrete Poisson equation produce circulant sub-matrices that are diagonalizable in a Fourier space. Consequently, a Fast Fourier Transform (FFT) method is used to solve the Poisson equation based on the explicit calculation and direct solution of a Schur Complement system for the independent 2D systems. More details about this method can be found in [32].

### 3. Adaptive Mesh Refinement

Mesh adaptation is accomplished by dividing or coarsening groups of cells following a refinement criteria, based on our physical understanding of the problem. Therefore, the AMR algorithm starts with an initial mesh and continuously refines certain regions by dividing a parent cell into four (two dimensions) or eight (three dimensions) children cells. While, in areas that are over resolved, the refinement process can be reversed by coarsening four or eight children cells into a single parent cell, following a quad/oct-tree scheme. Those processes are continuously performed, creating a suitable mesh for the solution of the vortical structures of the flow at each phase of the simulation. For algorithm convenience, the grid adaptation is constrained such as the cell resolution changes by only a factor of two between adjacent cells (see Figure 1) and the maximum level of refinement is established by a study of the Kolmogorov scales derived for the problem being considered.

#### 3.1. Mesh definition

In our software platform, a mesh is represented as an object composed of basic geometrical elements (vertices, faces and cells) and, which also describes the relationship between them. Hereafter the main data representing the basic elements of the mesh are described:

- *Vertex*: The class vertex consist in a vector with three spatial coordinates.
- *Face*: A face is a polygon and can be defined as an ordered set of integers corresponding to the indexes of its vertices.
- *Cell*: A cell is a polyhedron and is defined with the indexes of the face objects that form it. In this case no ordering is needed; a set of faces determines only one possible polyhedron.

For the face and cell objects, a list of its neighboring elements is stored. For example, for each cell are stored the global indexes of its neighbors. Those are at minimum the indexes of the 6 neighboring cells of the initial Cartesian mesh, but they may increase as the AMR process evolves.

### *3.2. Domain decomposition*

The mesh decomposition is derived from a partition of the cells adjacency graph, that is carried out by means of an external tool such as the ParMETIS library [33]. Apart than providing a good load balance, ParMETIS routines minimize the edge cuts, reducing the data exchange requirements in the simulation.

After the mesh partition, each parallel process deals with a subset of cells, faces and vertices that all together form a subdomain. These are referred to as owned elements of each type. Since the graph is defined by means of the cells adjacency its partition directly defines the owned cells of each parallel process.

The distribution of the faces and vertices is also based on the cells graph. However, on the subdomains boundary, where the elements are shared be-

tween processors, a rank criteria is used to establish the owner. For instance, when two neighbor cells are located on different subdomains, a common face is located on the border of the subdomain. This face is assigned to the processor with the higher rank.

The discrete operations are generally performed by means of scalar field defined over the elements of the mesh. Therefore, its distribution is determined by the distribution of the corresponding geometrical elements.

Note that, in the geometric and algebraic parallel operations, each parallel process may need elements owned by others. Therefore, a copy of the required elements, owned by other processors is attached. Those copies of external elements attached to each subdomain are referred as its halo. Its important to remark that any element of a halo is a copy, meaning that the original element is owned by another parallel process. Thus, if the original element changes in the owner parallel process, the copy stored in the halo must be updated before using it. Otherwise, the results of the sequential and parallel executions would differ.

Any mesh element is uniquely determined by its local identifier (lid), which refers to its position in a local storing container. However, the lid only identifies the element locally, i.e. different elements owned by different parallel processes may have the same lid. In order to globally determine each element, we use global identifiers (gid).

For each mesh element type, a topology object is created that contains the information of its corresponding local/global identifiers, that defines the domain decomposition (owned/halo elements) and, also the communication scheme required to update the halos.

### 3.3. Algorithm description

Writing a parallel AMR code for scientific computations is a laborious work. The overall AMR process, from the selection of the cells to be refined or coarsened, till the pre-processing of the resulting mesh, has been implemented in parallel, based on the standard domain decomposition (DD) method.

The major aspects regarding the parallelization, are the definition of a global order (i.e. the gids) for the discrete elements of the new mesh, and the operations performed to keep coherence on the subdomains borders.

The AMR algorithm inputs are the old mesh, i.e., the mesh being adapted, a list of global identifiers of cells and its corresponding level of refinement and, a tree data structure that keeps track of the cells decomposition, see Figure 2. The algorithm output is a new adapted mesh gathered into an unique data file from the submeshes generated by the different parallel processes, see Figure 3. Parallel IO operations are performed by means of the HDF5 library [34]. Finally, the new mesh partitioning is done with ParMETIS library, to achieve load-balance. This leads to a new partitioned mesh, that will be used for the next simulation step, see Figure 4.

The intermediate solutions are transmitted to the new mesh by means of interpolations based on the tree data structure. For the refinement process each solution data of the parent cell is set to its child cells, and for the coarsening process an average solution is given from the child cells to its corresponding parent cell.

A description of the code is presented in Algorithm 1, divided in four main steps.

*Algorithm 1.*

- Step1 (S1). *Read the old mesh and prepare the cells to be refined/coarsened.*

In this step, each processor uploads the old mesh from a data file and reads the list of cells with its corresponding level of refinement. If this level match with the cell level stored in the data file, the cells is not modified. However, if the level is higher, the cells will be refined. Differently than the refinement process, the coarsening is performed automatically (without requiring input data). The list of cells to be coarsened consist on the cells with level higher than 0 which are not in the input list of cells and neither are their neighbor cells. These operations are carried out according to the tree data structure. Moreover, communications are held between neighboring subdomains to ensure that the resolution between neighbor cells only changes by a factor of two.

- Step2 (S2). *Create a new mesh with the cells that are not modified.*

A mesh object is created with the elements that are not going to be refined/coarsened. A new index ordering is established for each processor, where collective communications are held in order to create an unique global index for each element. Storing containers are used to save the geometric information of the elements, meanwhile the information regarding the neighboring relations between elements is filled during the storing process. Communications are necessary to share the new global indexes of the elements on the boundary of the subdomains in order to complete the definition neighboring relations.

- Step3 (S3). *Create the new mesh elements and determine its corre-*

*sponding global indexes.* With the list of cells to be refined/coarsened, each processor will create the new vertices, cells and faces using the information of the old mesh and the tree data structure. In the case of the refinement process, new vertex are created by averaging the adjacent vertices coordinates of the parent cell. Then, the rest of the elements corresponding to the four/eight new cells are created. For the coarsening process, a new cell is created with all its corresponding elements. Finally, an index ordering is established for the new elements, taking into account the already existing elements from the Step 2. Collective communications are also required for this task. The new geometric elements created on the boundary of the subdomains need to be shared between parallel processes in order to ensure coherence and to complete the definition the neighboring relations

- Step4 (S4). *Create a data file with the new mesh.* Finally, all processors are synchronized and the data is gathered to create a HDF5 data file that contains the information of the new mesh that is used in the next step of the solution procedure. The creation of the HDF5 data file consist in two steps: first, collective communications are held in order to determine the global size of the data to be written. Second, each parallel process writes its respective information in a specific position determined by the global index of the elements. Hyperslab HDF5 functionalities are used in order to optimize this process.

### 3.4. *Parallel Performance*

The strong speed up of the AMR algorithm has been assessed for an initial Cartesian mesh of 1.1 million cells of which 1% are refined. In order to discard imbalance effects, the cells to be refined are homogeneously distributed throughout the domain. This test has been performed in an AMD Opteron based cluster (AMD Opteron 6272, 16-core processors at 2.1 GHz, 64 GB RAM linked with the infiniband QDR 4X network and a DDN S2A9900 with 80 disks of 1 Tb, obtaining a storage capacity of 64 Tb and writing at 2.6 Gbytes/s with a parallel file system called Lustre), engaging up to 256 CPU-cores. The relative weight as the number of CPU-cores grows, is shown in Table 1, for each of the four steps of Algorithm 1. Step 4, the creation of the final mesh file, is the part of the algorithm that further increases its cost, therefore, it becomes the main limitation for the speedup. The creation of this mesh file is managed by means of the HDF5 library, unfortunately, it does not provide good parallel performance on the writing operation. Contrary, the reading operation, performed in Step 1 with the same library, scales properly. In the Figure 5, is shown the strong speedup for both, the overall algorithm and also the algorithm obtained by discarding IO operations; i.e the original part of it. The parallel efficiency is clearly penalized by the IO operations. With 256 CPU-cores it reaches up to 90% for the rest of the code, but decreases down to 50% when the IO operations are included. Further work needs to be performed in order to extend the scalability to higher number of CPU-cores.

### 3.5. Conservation tests

In order to analyze the conservation properties for the proposed adaptive mesh refinement method, a Rankine vortex problem is chosen as test case. The Rankine vortex model is given by the combination of a rigid-body rotation within a core, a decay of angular velocity outside and zero mass flux at the boundaries. The tangential velocity,  $u_\theta$ , of a Rankine vortex with circulation,  $\Gamma$ , and radius,  $R$ , is given by

$$u_\theta(r) = \begin{cases} \Gamma r/2\pi R^2 & r \leq R, \\ \Gamma/2\pi r & r > R. \end{cases} \quad (8)$$

In particular, the Rankine vortex solved in this paper is placed in the center of a 3-D domain (1.0 x 1.0 x h), the initial tangential velocity reaches a maximum of 0.16  $m/s$  at radius  $R = 0.01 m$ , and circulation equals  $\Gamma = 0.032 \pi m^2/s$ . The density and viscosity of the fluid are  $\rho = 1.0 kg/m^3$  and  $\nu = 0.01 m^2/s$ , respectively. The domain is an adaptive mesh, with a refined area in the middle with 3 levels of refinement (Figure 6), where the refinement criteria is based on the vorticity field. All boundaries are slip walls.

Since there is no flow across the domain boundaries, if any difference exists between physical dissipation and the rate of change of total kinetic energy, it is due to the pressure error term, that arises from the special definition for the normal face velocity needed to exactly conserve mass in the collocated scheme [35, 28, 29].

For this test, a second-order spectro-consistent schemes on a collocated unstructured grid arrangement in a finite volume context was used. The

difference between rate of change of total kinetic energy,  $dk/dt = d(\frac{1}{2}\mathbf{u}\cdot\mathbf{u})/dt$ , and physical dissipation,  $-\nu\omega\cdot\omega$ , for an adaptive mesh was calculated at every time step using Eq. 9, and is compared with an uniform mesh with  $6.4\cdot 10^3$  cells. Results are plotted in Figure 7.

$$\begin{aligned}
& \sum_{c\in\Omega} \frac{d(\frac{1}{2}\mathbf{u}_c\cdot\mathbf{u}_c)}{dt} V_c + \sum_{f\in F(\partial\Omega)} \frac{1}{2}\mathbf{u}_a\cdot(2\phi_f - \phi_a)\hat{U}_f A_f = -\frac{1}{\rho} \sum_{f\in F(\partial\Omega)} p_f \hat{U}_f A_f \\
& -\frac{\delta t}{\rho^2} \sum_{c\in\Omega} p_c \sum_{f\in F(c)} \frac{1}{2} \left[ \frac{1}{V_c} \sum_{f\in F(c)} p_f \hat{\mathbf{n}}_f A_f + \frac{1}{V_{nb}} \sum_{f\in F(nb)} p_f \hat{\mathbf{n}}_f A_f \right] \cdot \hat{\mathbf{n}}_f A_f \\
& + \frac{\delta t}{\rho^2} \sum_{c\in\Omega} p_c \sum_{f\in F(c)} \left[ (p_{nb} - p_c) \frac{A_f}{\delta d_f} \right] + \nu \sum_{c\in\Omega} \mathbf{u}_c \cdot \sum_{f\in F(c)} (\mathbf{u}_{nb} - \mathbf{u}_c) \frac{A_f}{\delta d_f} \quad (9)
\end{aligned}$$

Results show a slightly perturbation when the mesh changes due to symmetry inconsistencies on the operators in the time integration. Thus, conservation is affected and dissipation is generated, but this phenomena is imperceptible for the global simulation as can be seen in Figure 7. Moreover, results show that the collocated scheme presents a decreasing difference of order  $10^{-9}$  for both uniform and AMR mesh.

However, it is important to notice that the kinetic energy error does not have a significant impact on the physics of the problem, because of the mesh size and time steps are small enough when direct numerical simulation (DNS) or large-eddy simulation (LES) are used to solve turbulent problems. This have been shown by Rodríguez et al. and Lehmkuhl et al. [15, 31, 27], which solve turbulent flows using the collocated scheme by means of DNS and LES techniques.

### 3.6. Refinement criteria

Physics-based mesh adaptivity requires criteria to establish measures that will indicate the refinement/coarsening process of the mesh. As mentioned earlier, we adopt in this work a mesh adaptivity criteria based on our physical understanding of the flow to identify the critical regions of the problem. For turbulent flows around bluff bodies considered here, measure of the residual velocity was calculated using the VMS scale separation theory, to focus on small scales range from the solution of the Navier-Stokes equations, and can be defined as:

$$\mathbf{u}' = \mathbf{u} - \bar{\mathbf{u}} \quad (10)$$

where  $\mathbf{u}'$  is the residual velocity,  $\mathbf{u}$  is the instantaneous velocity and  $\bar{\mathbf{u}}$  is the filtered velocity. In the present work to calculate the filtered velocity, we used a non-uniform Laplace filter based on a Gaussian filter that are normalized, conservative and also self-adjoint [26]. Thereby can be calculated on a general unstructured grid. The measure presented here is defined as:

$$\phi_c = \|\mathbf{u}'\| \quad (11)$$

where  $\phi_c \in R^m$  is the residual velocity magnitude (here m applies for the total number of control volumes (CV) of the discretized domain).

To identify the cells to be refined and coarsened, a global maximum value of the criteria is established.

$$\phi_{max^o} = \max[\phi_c] \quad (12)$$

Moreover, the global maximum value is averaged in time to keep a smooth adaptation effect.

$$\phi_{max^n} = [\phi_{max^n}(t) + \phi_{max^n}(\Delta t)] \frac{1}{t + \Delta t} \quad (13)$$

Then, an average value is calculated with the cell values above 10% of the global maximum value to avoid the cells where the residual velocity is near to zero.

$$\phi_{avg^n} = \frac{\sum_0^m f(\phi_c) \cdot \phi_c}{\sum_0^m f(\phi_c)} \quad (14)$$

where,  $f(\phi_c)$  is given by,

$$f(\phi_c) = \begin{cases} 1 & \phi_c \geq 0.1(\phi_{max^n}), \\ 0 & \phi_c < 0.1(\phi_{max^n}). \end{cases} \quad (15)$$

Therefore, the threshold can be defined as

$$\epsilon_c = \frac{\phi_{avg^n}}{\phi_c} \quad (16)$$

With this parameter, the cells to be refined are those with  $\epsilon_c \geq 2$ . Based on our experience, a more conservative approach for the coarsen process was adopted, for which the coarsening limit to the cells with  $\epsilon_c < 2$  and those who are not neighbors with the cells marked to be refined.

In Figure 8, an example of the use of this criteria for the square cylinder problem at  $Re=22000$  is shown. Most of the vortical structures are captured in refined cells whether near the object and in the wake region. Other measures can be considered, i.e. the vorticity field. But, as can be seen in Figure

9, the results are not very promising because this field is flow-dependent, therefore there are important zones that are not refined even if a vortical structure is present. Moreover, the use of this field requires continual tuning depending on the flow problem, as studied by S.J. Kamkar [11]

The residual velocity criteria presents a better behavior and the threshold can be established in a general way to resolve basic turbulent problems without user intervention of the refinement process. This criteria, with the parameters established in this section, has been applied in other turbulent problems, using an immersed boundary technique, like the flow over a circular cylinder at  $Re=3900$  and the flow past a NACA 0012 at Reynolds number  $Re = 5 \cdot 10^5$  with a low angle-of-attack  $AOA = 5^\circ$ .

The criteria has worked well for both problems where the flow over a circular cylinder exhibits a transition to turbulence in separated shear layers, and interactions between shear-layers and the turbulent wake (See Figure 10). Meanwhile, the flow past a NACA 0012 presents a laminar separation, transition to turbulence in the separated shear-layer and a laminar separation bubble when the flow reattaches to the airfoil surface. (See Figure 11).

#### **4. Numerical results of turbulent flows using AMR-LES**

As has been show, the refinement criteria developed seems to refine the areas needed to solve the smallest flow structures on most of the turbulent problems around bluff bodies. Moreover, the AMR mesh has shown to be able to preserves well the kinetic energy balance to ensure the conservation properties to solve turbulent problems. After these results, it would be desirable to test how our methodology deals with the turbulent flow around bluff

bodies at higher Reynolds numbers. Hereafter two cases have been studied: (i) the flow around a square cylinder at  $Re=22000$  and (ii) the flow around two side-by-side square cylinders at  $Re=21000$ . In both cases, numerical results have been compared with experimental and numerical results from the literature.

#### *4.1. Flow around a square cylinder at $Re=22000$*

Numerical simulations of the flow around a square cylinder are performed at  $Re = 22000$  [18, 19, 20, 21, 22], where Reynolds number is defined in terms of the free-stream velocity  $U$  and the square length  $L$ . The results presented in this paper have been obtained using a computational domain of dimensions  $[-5.5L, 14.5L]$ ;  $[-7L, 7L]$ ;  $[0, 4L]$  in the stream-, cross- and span-wise directions respectively, where the square cylinder is located at  $x = 0$ ,  $y = 0$  (See Figure 12).

The governing equations are solved on an adaptive mesh generated from the extrusion around the axis of a two-dimensional grid in a  $(x,y)$  plane. The boundary conditions at the inflow consist of a uniform velocity  $(u,v,w)=(1,0,0)$ . A pressure based condition is used at the outlet boundary for the downstream. At the cylinder surface, no-slip conditions are prescribed. As for the span-wise direction, periodic boundary conditions are imposed. The use of an adaptive mesh for the plane, with four mesh levels, has allowed to cluster more control volumes around the cylinder surface and in the near wake. For this case, the mesh adaptivity is performed after every 0.13 computational time units, this was set in an empirical way related to the phenomenon of vortex shedding for bluff bodies, where a fraction of the non-dimensional shedding frequency (Strouhal number) was used. At the end of the simula-

tion, the total mesh has around (1.2MCVs) and 16 planes, for the periodic direction, have been also considered for this simulation.

For obtaining the numerical results presented here, simulations have been started from an initial homogeneous flow field. Then, advanced in time until statistical stationary flow conditions have been achieved. Results have been obtained based on the integration of instantaneous data over a sufficiently time period. Some instantaneous snapshots are depicted in Figure 13. Vorticity structures in the near wake obtained with the adaptive grid are plotted in Figure 13 (left) and the computational grid for that time step is plotted in Figure 13 (right).

The resulting time-averaged flow parameters are summarized in Table 2. The mean Strouhal number, the mean drag coefficient ( $C_d$ ), the mean lift coefficient ( $C_l$ ) and the rms fluctuations of  $C_d$  and  $C_l$  are presented (see Figure 14, for the time variation of the drag and lift coefficients). For comparison, experimental and numerical (from DNS and LES) results from the literature are also given [18, 19, 20, 21, 22]. As can be observed, the AMR-LES predicts the computed flow parameters in good agreement with the ones in literature, using the refinement criteria mentioned before. Although, the results show slightly differences for the drag and lift coefficient, but the Strouhal number is in fair agreement with the literature results (see Figure 15, for the power spectral density graph that shows a peak at  $f=0.133$ ). This can be related to the refinement criteria operation, that establish a big refinement zone with the maximum level possible near the body and in the wake region.

In order to gain more insight into the behavior of the AMR-LES, the stream-wise and cross-streamwise velocity profile and its fluctuations are

plotted at different locations near the cylinder (at  $x/D = -0.5$ ,  $x/D = -0.25$ ,  $x/D = 0.5$ ,  $x/D = 1.5$  and  $x/D = 3$ , Figure 16 and 17). For comparison the experimental results from Lyn, D.A. et al. [18, 19] are also included. As can be seen, the AMR-LES results present good agreement with the experimental data. The first-order statistics are quite well predicted near the body and in the wake region. But, slightly differences can be observed at the velocity fluctuations near the body, therefore a finest grid is required for a better prediction. It is also interesting to observe some minor discrepancies in the wake region ( $x/D = 1.5$ ), whereas the transition takes place in the separate shear layers on the side of the cylinder.

These good results can be attributed to the use of a conservative discretization of the convective and diffusive operators, as they preserve the kinetic energy balance. It should be pointed out that the refinement criteria presented here, shows a good behavior following the small vortical structures, given the spatial resolution needed to achieve better numerical results for the LES model. It also can be noticed that the impact on the results of the dynamic procedure within the VMS-LES approach is rather small, and it is attributed to separation of the scales, where the turbulent viscosity modeling is acting directly only on the resolved small scales. With this strategy, SGS viscosity is only introduced in the unresolved small scales improving the behavior of the LES model.

#### *4.2. Flow around two side-by-side square cylinders at $Re=21000$*

Numerical simulations of the flow around two side-by-side square cylinders at Reynolds number  $Re = 21000$  [23, 24] (Reynolds number is defined in terms of the free-stream velocity  $U_{ref}$  and the square length  $L$ ) and

$g^*=g/L=6$  (gap ratio is defined as  $g^*=g/L$ , where  $g$  represents the spacing between the square cylinder surfaces) have been carried out. Solutions are obtained in a computational domain of dimensions  $41L \times 28L \times 4L$  where the first square cylinder is located at  $x = 10, y = 10$ , and the second square cylinder is located at  $x = 10, y = 17$  (see Figure 18). All coordinates are referred to body axes unless remarked. The  $x$  axis is stream-wise,  $y$  is the cross-wise and  $z$  is span-wise direction. The boundary conditions at the inflow consist of a uniform velocity  $(u,v,w)=(1,0,0)$ . At the outflow boundary, a pressure-based condition is imposed. No-slip conditions on the square surfaces are prescribed. Periodic boundary conditions are used in the spanwise direction. The characteristics for the adaptive method are the same used for the single square cylinder mentioned in the last section. At the end of the simulation the total mesh has around (2.06MCVs) and 16 planes, for the periodic direction, have been also considered for this simulation. In Figure 19, a detail of the mesh of the plane in the region of the square cylinders is depicted. As can be seen, control volumes have been clustered near the cylinders surfaces, as well as in the wake region.

LES computation have been performed with the VMS-WALE model, and the results show that the Strouhal number obtained is 13.3, which is in good agreement with the value measured in experiments and numerical studies. As can be seen, the flow structure behind each square cylinder is independent. There are not close interactions between the shear layers and the flow separation that occurs near the surfaces of the cylinders, thus the vortex shedding frequency is similar to the result for the singular square cylinder. A complete comparison of aerodynamic coefficients against experimental and

numerical data [23, 24] is depicted in Table 3. A good agreement between the calculated results and the experimental/numerical data has been achieved for this gap. It should be noted, that the flow structure exhibits anti-phase vortex shedding [23, 24], where two distinct vortex streets separate from the cylinders and vortices can stably proceed to the far downstream wake zone, see Figure 19. It also can be noticed, that the refinement criteria and the adaptive parameters used for the single square cylinder have been used for the two side-by-side square cylinders, where most of the flow features were captured and control volumes were clustered in the regions where the grid must be dense enough to capture all the flow scales.

## 5. Concluding remarks

A parallel adaptive mesh refinement algorithm has been presented. It has been developed to optimize the LES simulation of turbulent flows. The underlying discretization of the Navier-Stokes equations has been arranged by means of a non dissipative symmetry preserving method. And a new refinement criteria tightly coupled with the LES model has been developed.

The work presented is focused on three main aspects. First of all we assessed the conservation properties of the discretization for the meshes resulting from the AMR process, characterized by the transitions between zones with different level of refinement. Tests performed for a Rankine vortex problem, show an almost negligible kinetic energy error without any noticeable impact on the physics of the problem.

Secondly, a new mesh adaptivity criterion has been developed. It is based on the VMS scales separation theory, which allows to better discriminate the

unresolved scales of motion. This methodology, has been successfully tested for different bluff body problems without particular tuning for any of them.

Third, the overall AMR algorithm has been developed to be executed in parallel. In particular, the algorithm shows a good strong speedup with up to 256 CPU-cores, on the refinement of an homogeneously distributed group of cells of a Cartesian mesh. We have assessed that the principal limitation for the parallel performance are the IO operations. The speedup of the rest of the algorithm reaches 90% on 256 CPU-cores. Further work needs to be done in order to extend the parallelization to larger numbers of CPU-cores.

Finally, the AMR-LES method has been applied on the simulation of the turbulent flow around a square cylinder at  $Re = 22000$  and turbulent flow around two side-by-side square cylinders at  $Re=21000$ . Main features of the flow (flow separation, vortex shedding, turbulent wake, etc) were successfully captured. Numerical results are in good agreement with previous references demonstrating the robustness of the presented approach.

## Acknowledgments

The work has been partially financially supported by the Ministerio de Ciencia e Innovación, Spain (ENE2010-17801). The authors thankfully acknowledge this institution.

## References

- [1] M. J. Berger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, vol. 53, pp. 484 – 512, 1984

- [2] M. J. Berger and R. J. LeVeque. An adaptive cartesian mesh algorithm for the euler equations in arbitrary geometries. *AIAA*,, 1989
- [3] K. G. Powell, P. L. Roe, and J. Quirk. Adaptive-mesh algorithms for computational fluid dynamics. *In M. Y. Hussaini, A. Kumar, and M. D. Salas, editors, Algorithmic Trends in Computational Fluid Dynamics*, vol. 53, pp. 303 – 337, 1993
- [4] S. Prudhomme, J.T. Oden. A posteriori error estimation and error control for finite element approximations of the time-dependent Navier-Stokes equations. *Finite Elements in Analysis and Design*, pp. 247 – 262, 1999
- [5] J. Hoffman. Computation of mean drag for bluff body problems using adaptive DNS/LES. *SIAM J. Sci. Comput.*, vol. 27, pp. 184 – 207, 2005
- [6] J. Hoffman, C. Johnson. Stability of the dual Navier–Stokes equations and efficient computation of mean output in turbulent flow using adaptive DNS/LES. *Comput. Methods Appl. Mech. Engrg*, vol. 195, pp. 1709 – 1721, 2006
- [7] S. Berrone, M. Marro. Space–time adaptive simulations for unsteady Navier–Stokes problems. *Computer and Fluids*, vol. 38, pp. 1132 – 1144, 2009
- [8] J.P.P. Magalhães, D.M.S. Albuquerque, J.M.C. Pereira, J.C.F. Pereira. Adaptive Mesh Finite-Volume Calculation of 2D Lid-Cavity Corner Vortices. *Journal of Computational Physics*, 2013

- [9] Xinfeng Gao, Clinton P.T. Groth. A parallel solution – adaptive method for three-dimensional turbulent non-premixed combusting flows. *Journal of Computational Physics*, vol. 229, pp. 3250 – 3275, 2010
- [10] Dumbser Michael, Zanotti Olindo, Hidalgo Arturo, Dinshaw Balsara. ADER-WENO finite volume schemes with space-time adaptive mesh refinement. *Journal of Computational Physics*, vol. 248, pp. 257–286, 2013
- [11] S.J. Kamkar, A.M. Wissink, V. Sankaran, A. Jameson. Feature-driven Cartesian adaptive mesh refinement for vortex-dominated flows. *Journal of Computational Physics*, vol. 230, pp. 6271 – 6298, 2011
- [12] O. Lehmkuhl, C.D. Perez-Segarra, R. Borrell, M. Soria, and A. Oliva. TERMOFLUIDS: A new parallel unstructured CFD code for the simulation of turbulent industrial problems on low cost PC cluster. *In Proceedings of the Parallel CFD 2007 Conference*, pp. 1 – 8, 2007
- [13] R.W.C.P. Verstappen , A.E.P. Veldman. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, vol. 187, pp. 343 – 368, 2003
- [14] O. Lehmkuhl, I. Rodríguez, R. Borrell and A. Oliva. Low-frequency unsteadiness in the vortex formation region of a circular cylinder. *Phys. Fluids*, vol. 25(8), 2013
- [15] I. Rodríguez, R. Borrell, O. Lehmkuhl, C.D. Perez-Segarra, and A. Oliva. Direct Numerical Simulation of the Flow Over a Sphere at  $Re = 3700$ . *Journal of Fluid Mechanics*, vol. 679, pp. 263 – 287, 2011

- [16] F. Nicoud and F. Ducros. Subgrid-scale stress modeling based on the square of the velocity gradient tensor. *Flow, Turbulence and Combustion*, vol. 62, pp. 183 – 200, 1999
- [17] T.J.R. Hughes, L. Mazzei, and K.E. Jansen. Large eddy simulation and the variational multiscale method. *Computing and Visualization in Science*, vol. 3, pp. 47 – 59, 2000
- [18] Lyn, D.A., Einav, S., Rodi, W. and Park, J.H. A laser-Doppler velocimetry study of ensemble-averaged characteristics of the turbulent near wake of a square cylinder. *Rept. SFB 210 /E/100*, (1994)
- [19] Lyn, D.A. and Rodi, W. The flapping shear layer formed by flow separation from the forward corner of a square cylinder. 1994 .
- [20] Sohankar, A., Davidson, L. Large Eddy Simulation of Flow Past a Square Cylinder: Comparison of Different Subgrid Scale Models. *J. Fluids Eng.*, vol. 122, pp. 39 – 47, 2000
- [21] R. W. C. P. Verstappen, A. E. P. Veldman. Fourth-Order DNS of Flow Past a Square Cylinder: First Results. *Direct and Large-Eddy Simulation II ERCOFTAC Series*, vol. 5, pp. 381 – 384, 1997
- [22] S.C. Luo, Md.G. Yazdani, Y.T. Chew, T.S. Lee. Effects of incidence and afterbody shape on flow past bluff cylinders. *J. Wind Eng. Ind. Aerodyn*, vol. 53, pp. 375 – 399, 1994
- [23] Shun C. Yen and Jung H. Liu. Wake flow behind two side-by-side square cylinders. *International Journal of Heat and Fluid Flow*, vol. 32, pp. 41 – 51, 2011

- [24] M. Mirzaei and A. Sohankar. Numerical study of convective heat transfer and fluid flow around two side by side square cylinders using  $k-w-v^2-f$  turbulence model. *Heat Mass Transfer*, vol. 49, pp. 1755 – 1769, 2013
- [25] Sagaut P, Germano M. Large eddy simulation for incompressible flows. *Springer-Verlag*, , 2001
- [26] Vreman AW. The adjoint filter operator in large-eddy simulation of turbulent flow. *Phys Fluids*, vol. 16(6), , 2004
- [27] O. Lehmkuhl, I. Rodríguez, A. Baez, A. Oliva, C.D. Perez-Segarra. On the large eddy simulations for the flow around aerodynamic profiles using unstructured grids. *Comput Fluids*, vol. 84, pp. 176 – 189, 2013
- [28] L. Jofre, O. Lehmkuhl, J. Ventosa, F. X. Trias, A. Oliva. Conservation Properties of Unstructured Finite-Volume Mesh Schemes for the Navier-Stokes Equations. *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, vol. 65(1), pp. 53 – 79, 2013
- [29] F.X. Trias, O. Lehmkuhl, A. Oliva, C.D. Perez-Segarra, R.W.C.P. Verstappen. Symmetry-preserving discretization of Navier Stokes equations on collocated unstructured grids. *Journal of Computational Physics*, vol. 258, pp. 246 – 267, 2013
- [30] F. X. Trias and O. Lehmkuhl. A self-adaptive strategy for the time integration of Navier-Stokes equations. *Numerical Heat Transfer, Part B: Fundamentals*, vol. 60(2), pp. 116 – 134, 2011

- [31] Rodríguez I., Lehmkuhl O., Borrell R., Oliva A. Flow dynamics in the wake of a sphere at sub-critical Reynolds numbers. *Comput Fluids*, vol. 80, pp. 233 – 243, 2013
- [32] Borrell R., Lehmkuhl O., Trias F., Oliva A. Parallel direct Poisson solver for discretisations with one Fourier diagonalisable direction. *Comput Phys*, vol. 230(12), pp. 4723 – 4741, 2011
- [33] K. Schloegel, G. Karypis, and V. Kumar. ParMETIS, Parallel graph partitioning and sparse matrix ordering library. *Journal of Computational Physics*, 2009
- [34] The HDF Group. Hierarchical data format version 5. 2000-2010. <http://www.hdfgroup.org/HDF5>.
- [35] F.N. Felten and T.S. Lund. Kinetic energy conservation issues associated with the collocated mesh scheme for incompressible flow. *Journal of Computational Physics*, vol. 215(2), pp. 465 – 484, 2006

## Figures

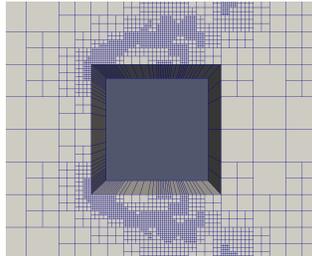


Figure 1: AMR cell-based scheme

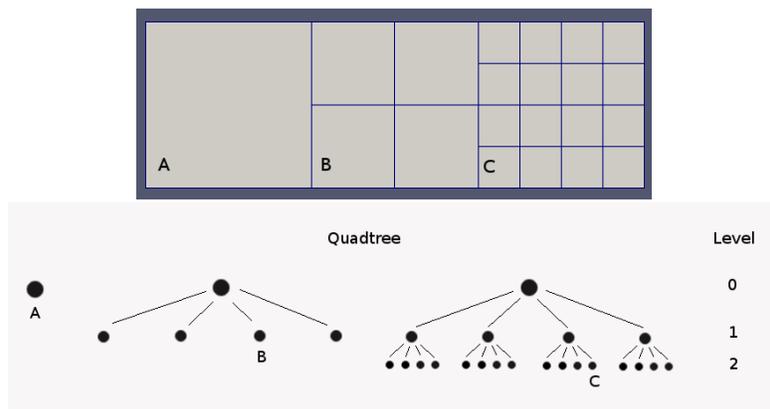


Figure 2: AMR mesh showing various levels of refinement and its corresponding quadtree data structure.

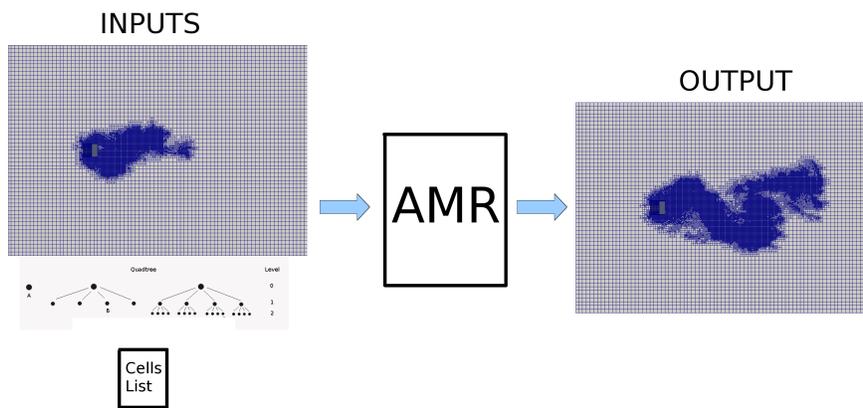


Figure 3: Schematic figure for the inputs/outputs of the algorithm

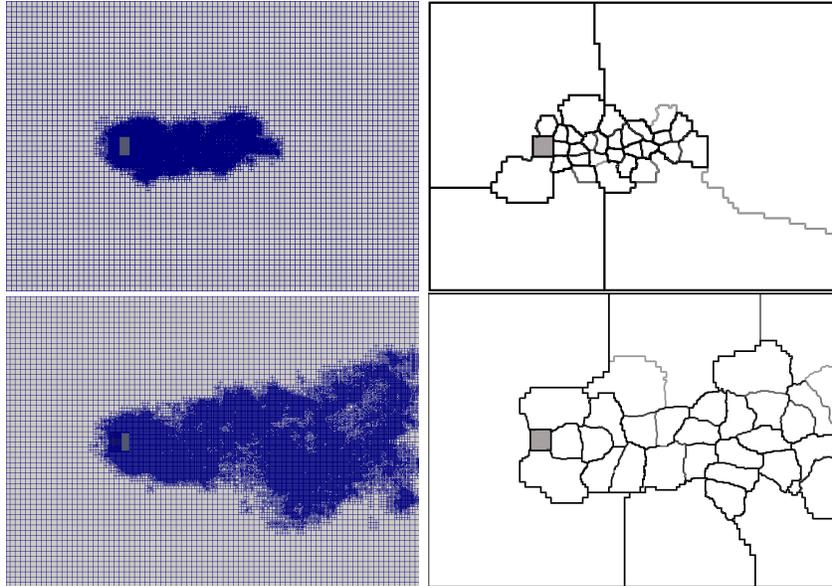


Figure 4: Parallel partitioning, among 32 processors, for different meshes in the solution of a square cylinder problem. The dark lines indicate partition boundaries.

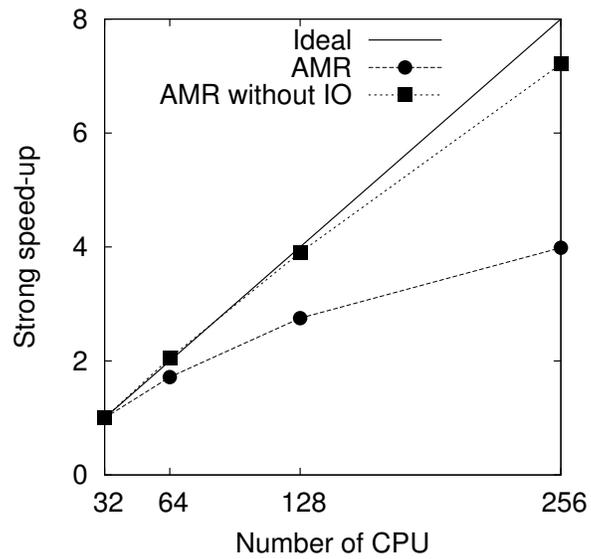


Figure 5: Strong speedup of the AMR algorithm with and without IO operations.

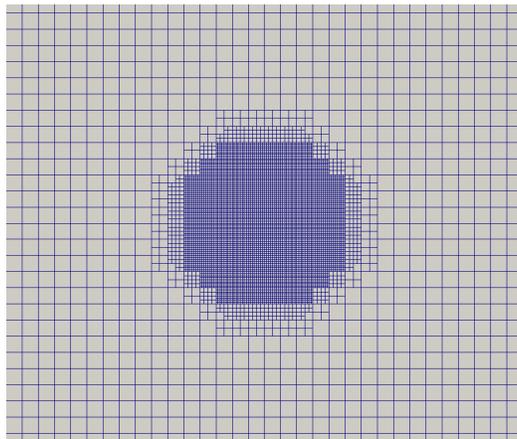


Figure 6: Illustration of the computational domain with 3 levels of refinement.

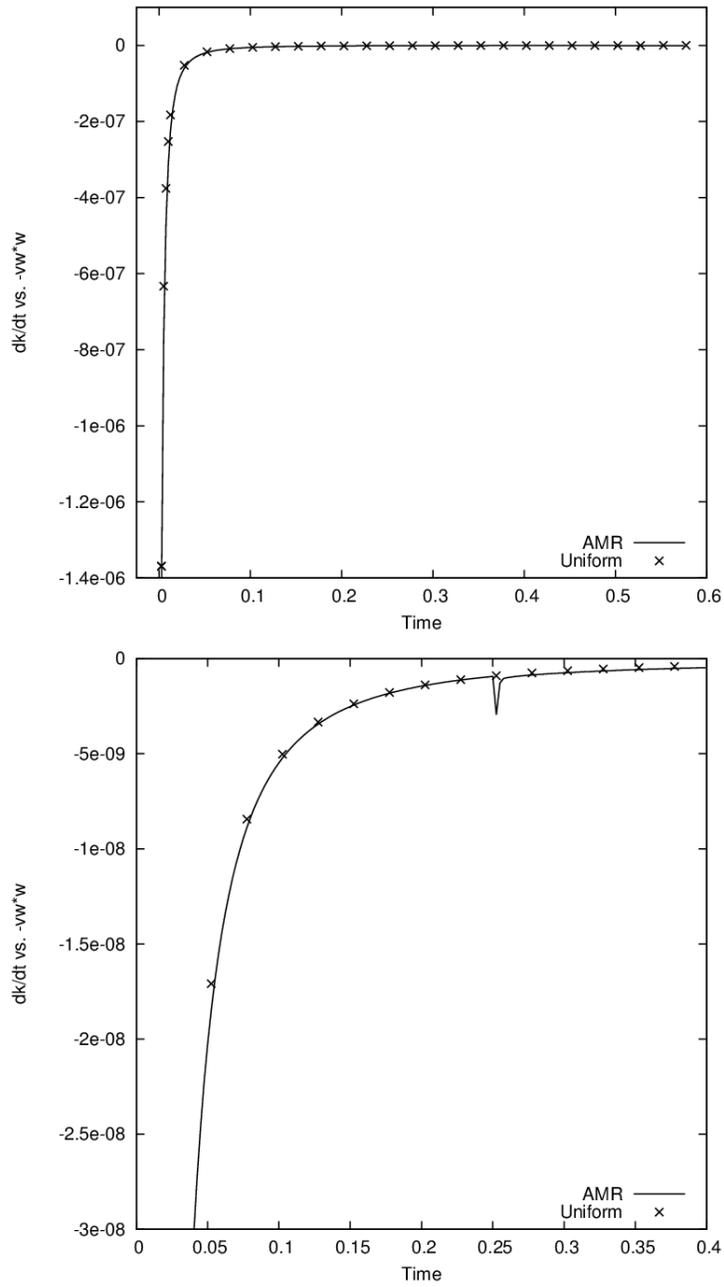


Figure 7: Difference between kinetic energy rate of change and physical dissipation using collocated scheme versus time with  $\nu = 0.01$  (left) Entire simulation (right) Zoom in when the mesh has changed.

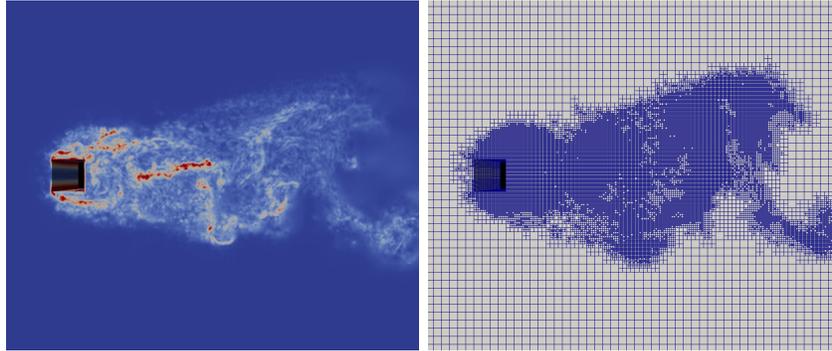


Figure 8: Vorticity field of the square cylinder problem at  $Re=22000$ . Residual velocity was used as refinement criteria.

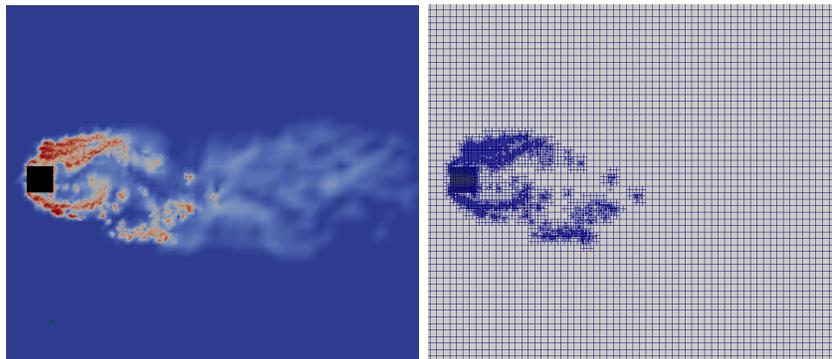


Figure 9: Vorticity field of the square cylinder problem at  $Re=22000$ . Vorticity was used as refinement criteria.

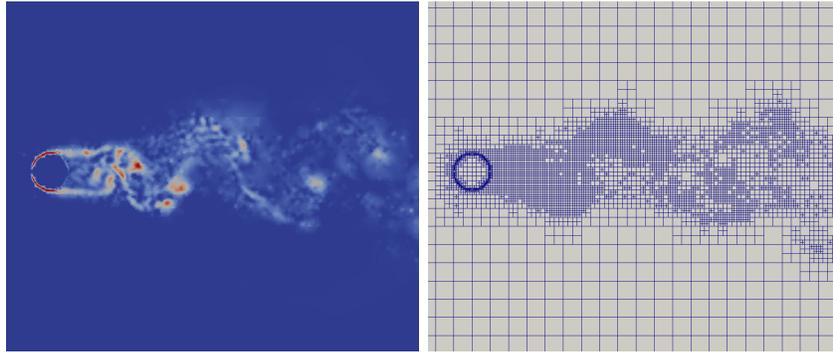


Figure 10: Illustration of vortical structures for LES of turbulent flow using residual velocity as refinement criteria: Circular cylinder at  $Re=3900$ .

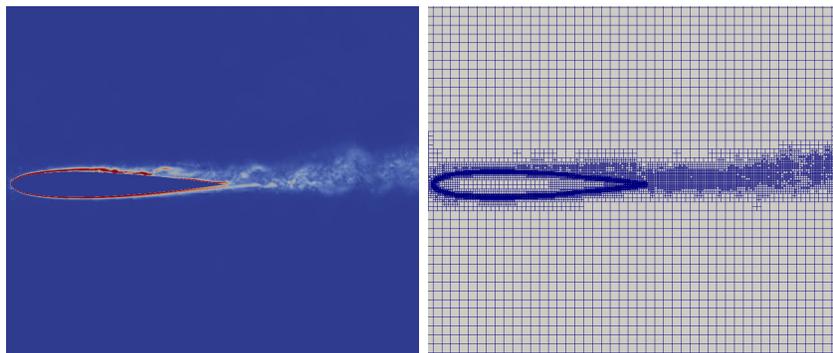


Figure 11: Illustration of vortical structures for LES of turbulent flow using residual velocity as refinement criteria: NACA0012 at  $Re = 5 \cdot 10^5$  and  $AOA = 5^\circ$ .

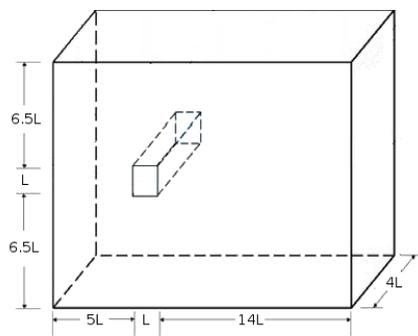


Figure 12: Computational domain for turbulent flow around a square cylinder at  $Re=22000$

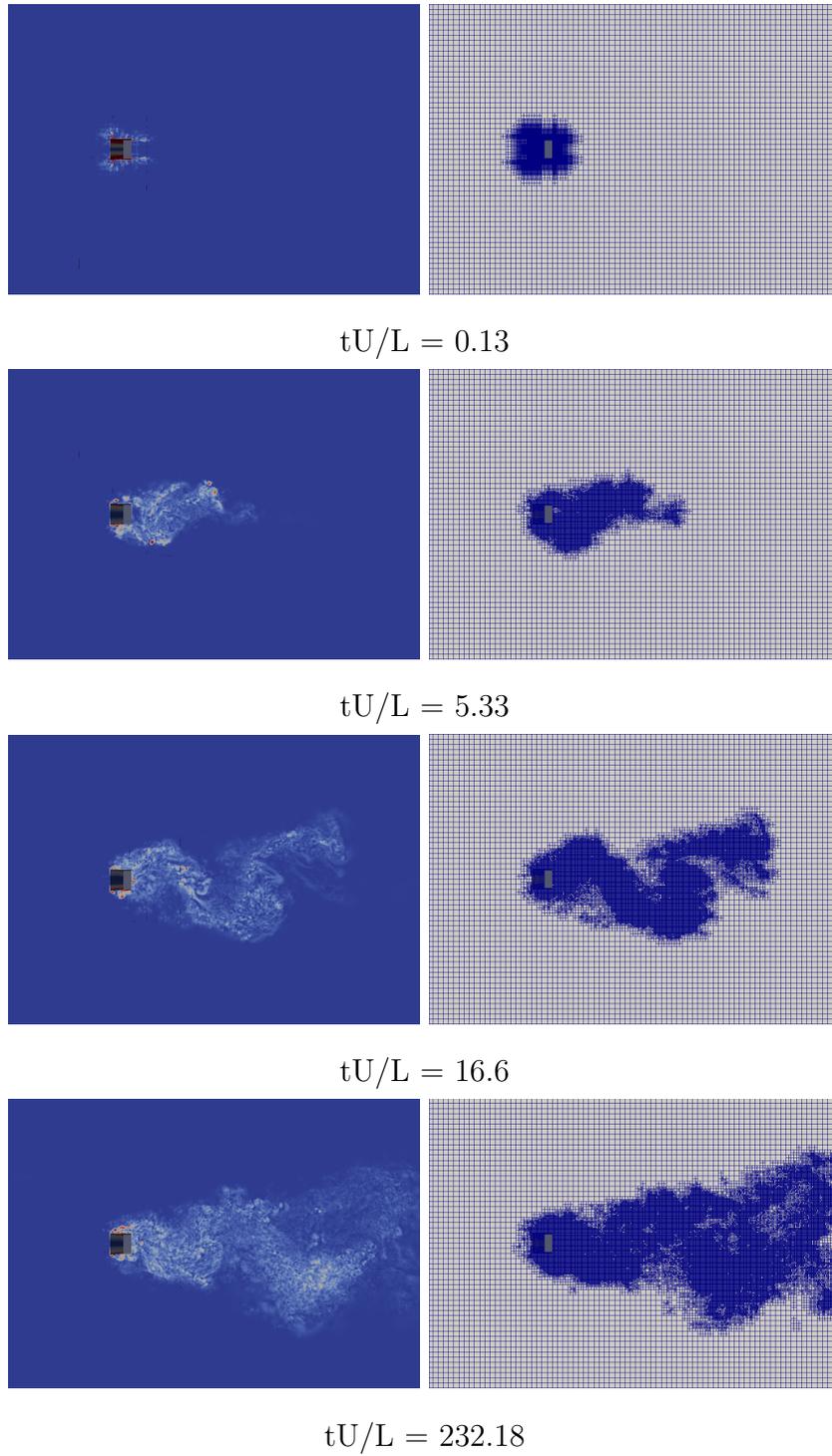


Figure 13: Illustration LES of turbulent flow around a square cylinder (left) Vorticity structures (right) computational grid.

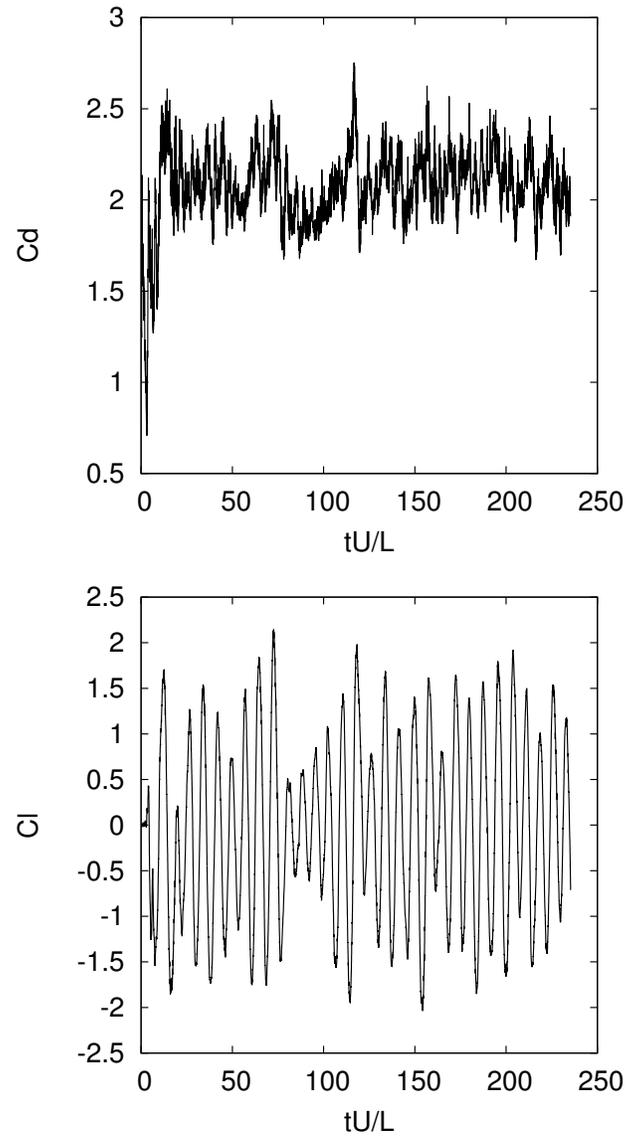


Figure 14: Time variation of (top) Drag and (bottom) Lift coefficients

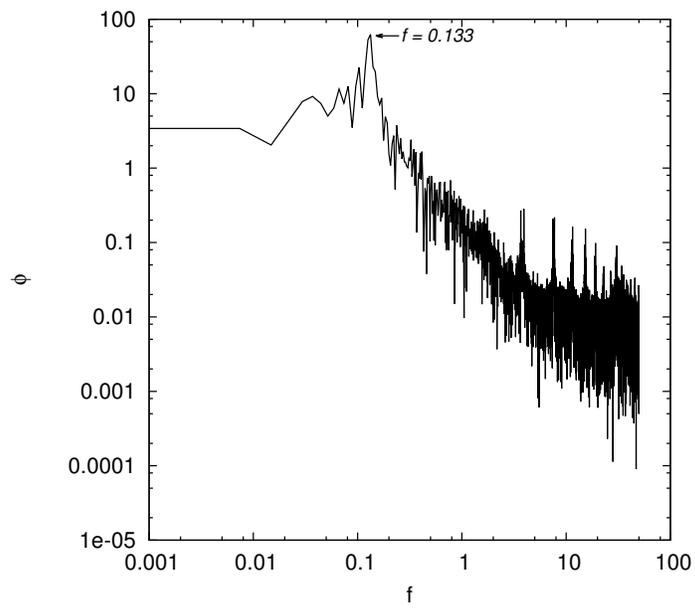


Figure 15: Power spectral density graph shows a peak at  $f=0.133$

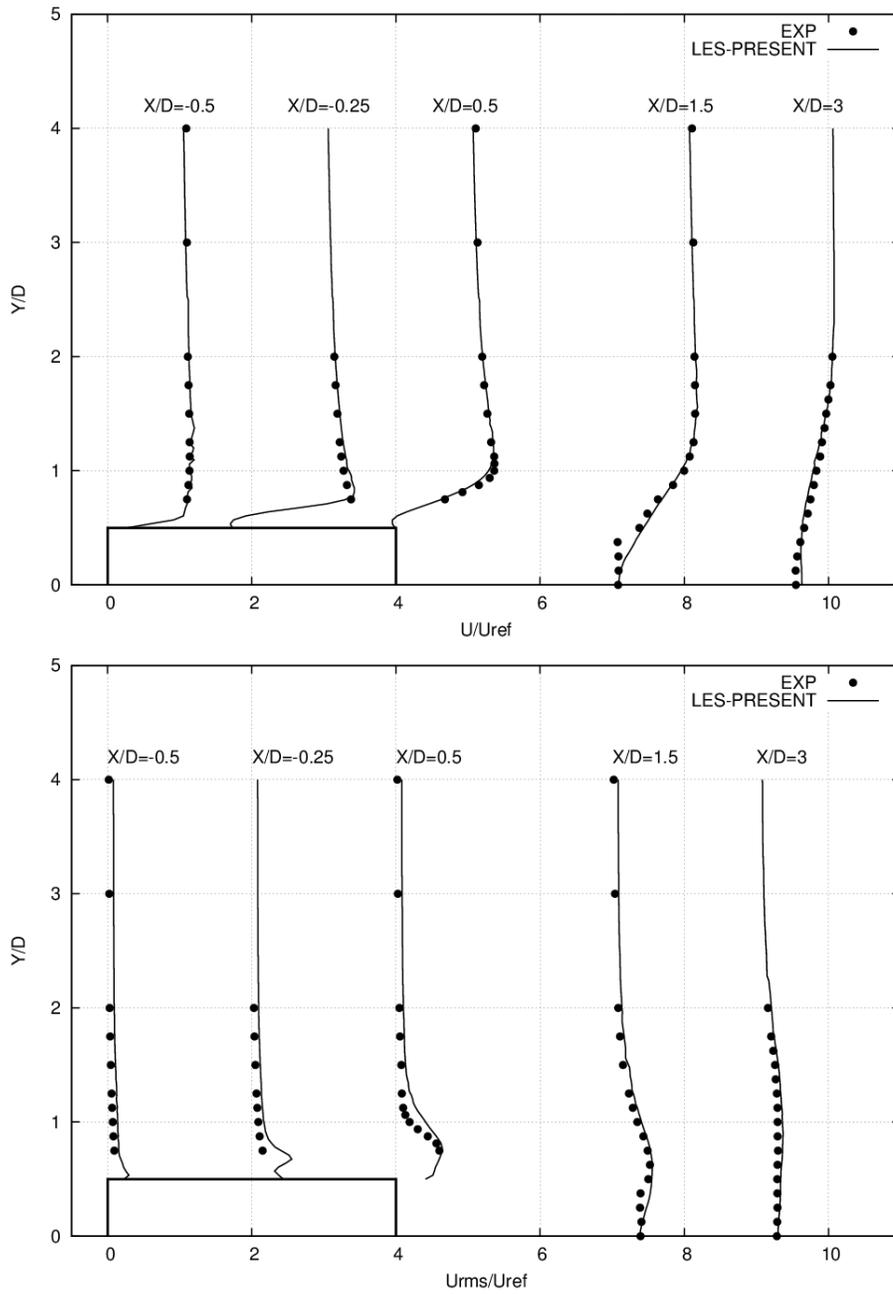


Figure 16: Comparison with experimental data. (up) Average streamwise velocity (down) Root mean square streamwise velocity fluctuations

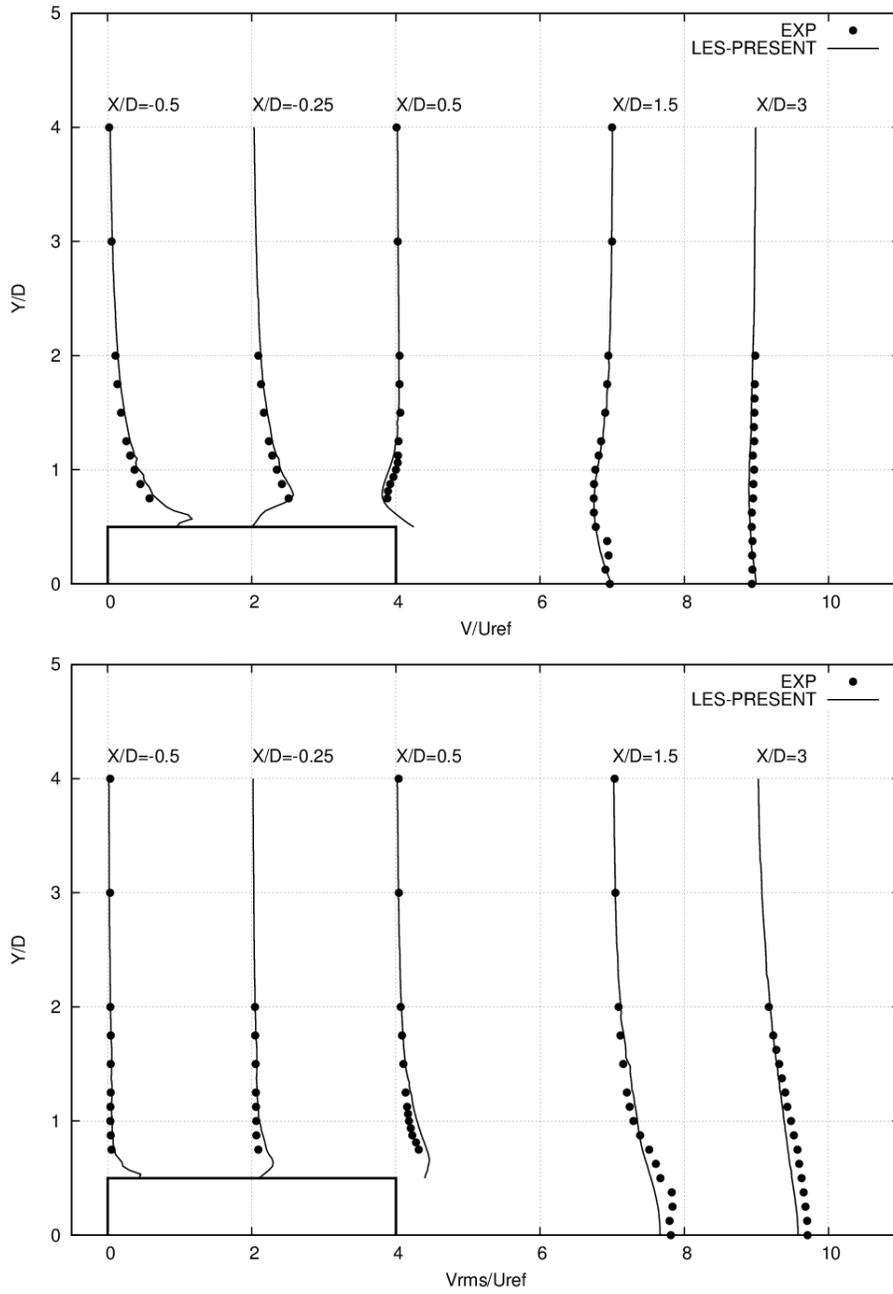


Figure 17: Comparison with experimental data. (up) Average cross-streamwise velocity  
 (down) Root mean square cross-streamwise velocity fluctuations

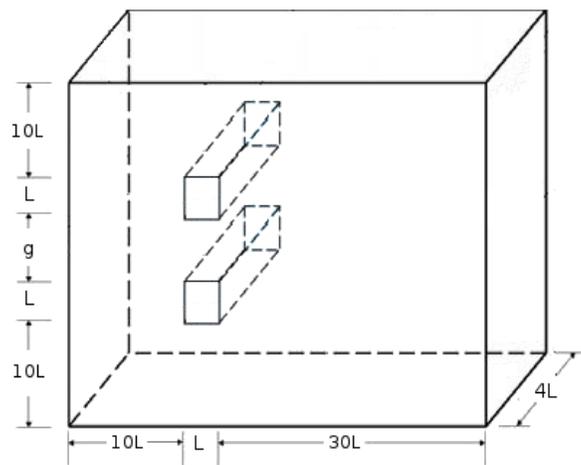


Figure 18: Computational domain for turbulent flow around two side-by-side square cylinders at  $Re=21000$

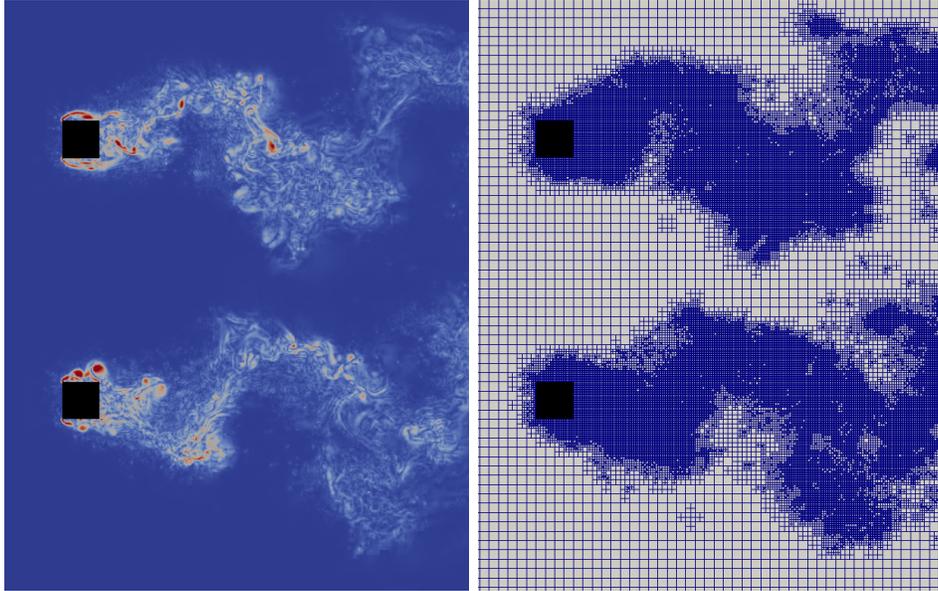


Figure 19: Illustration LES of turbulent flow around two side-by-side square cylinders (left) Vorticity structures (right) computational grid.

## Tables

Number of CPU	32	64	128	256
Step 1	37.9	32.5	25.3	21.7
Step 2	18.8	15.6	13.0	12.5
Step 3	23.3	18.8	13.5	9.9
Step 4	20.0	33.1	48.2	55.9

Table 1: Relative weight (%) of each of the four steps of Algorithm 1 for different number of CPU-cores.

	$Cd_{mean}$	$Cl_{mean}$	$Cd_{rms}$	$Cl_{rms}$	$St$
Present work	2.024	0.016	0.154	1.036	0.133
Sohankar (LES) [20]	2.03-2.32	-	0.16-0.2	1.23-1.54	0.126-0.132
Verstappen (DNS) [21]	2.1	0.005	0.21	1.22	0.133
Lyn (Exp) [18, 19]	2.1	-	-	-	0.132
Luo (Exp) [22]	2.2	-	0.18	1.2	0.13

Table 2: Time-averaged flow parameters for flow around a square cylinder at  $Re=22000$ .

	$Cd1_{mean}$	$Cl1_{mean}$	$Cd2_{mean}$	$Cl2_{mean}$	$St$
Present work	2.001	-0.085	2.056	0.069	0.13
Mirzaei (Num) [24]	2.117	-0.075	2.117	-0.075	0.13
Yen (Exp) [23]	2.08	-	2.08	-	0.13

Table 3: Time-averaged flow parameters for flow around two side-by-side square cylinders at  $Re=21000$ , where 1 identifies the upper cylinder and 2 the lower cylinder.