

TOWARDS NUMERICAL SIMULATION OF OFFSHORE WIND TURBINES USING ANISOTROPIC MESH ADAPTATION

L. Douteau^{1,2}, L. Silva¹, H. Digonnet¹, T. Coupez¹, D. Le Touzé², J.-C.
Gilloteaux²

¹ High Performance Computing Institute (ICI)
Centrale Nantes, France
Louis.Douteau@ec-nantes.fr, ici.ec-nantes.fr

² Research Laboratory in Hydrodynamics, Energy and Atmospheric Environment (LHEEA)
Centrale Nantes, France
lheea.ec-nantes.fr

Key words: Floating wind turbines, Anisotropic mesh adaptation, Multiphase Navier-Stokes, Variational MultiScale, High performance computing

Abstract. Offshore wind turbines are currently increasing the potential of wind energy, and numerical simulation is a way to help this industry to reach maturity. In the context of floating wind energy, predicting the loads applied on structures and their response is essential. Those data will enable an optimization of floaters dimensioning, necessary for CAPEX reduction.

As the simulation of floating wind turbines requires the representation of both complex geometries and phenomena, several alternative techniques have been developed. The wake generated by the rotor can be modeled using methodologies inherited from onshore wind turbines simulation, and coupled with a hydrodynamic code. However those simplified methods have been primarily developed for wake study, and thus have varying precision for loads estimation.

This work proposes a methodology for the simulation of a single or several turbines with an exact representation of the geometries involved, targeting an accurate evaluation of loads. The software library used is ICI-tech, developed at the High Performance Computing Institute (ICI) of Centrale Nantes. A monolithic approach is applied on a single computational mesh, where all the different phases are defined through level-set functions. The Navier-Stokes (NS) equations are solved in the Variational MultiScale (VMS) formalism using stabilized finite elements. This approach, coupled with an automatic and anisotropic adaptation procedure, guarantees the good representation of the geometries immersed. The automatic adaptation refines the mesh only in interest zones, allowing the simulation of phenomena with very different orders of magnitude, e.g. aerodynamics around blades and waves propagation. The reduction of the number of points in the mesh and the massive parallelization of the code are also necessary for wind turbine simulation.

This has been realized within the EOS project scope, funded by WeAMEC (West Atlantic Marine Energy Center, https://www.weamec.fr/en/blog/record_project/eos/).

1 Introduction

The development of renewable energies has significantly accelerated over the last few years. Wind energy is particularly interesting to that extend with massive amount of knowledge gathered, as wind farms have been implanted onshore for more than a decade. However, the future of wind energy seems to be located offshore. Ground bound wind farms have been producing energy for a decade, especially in North Sea, but only few places are eligible to the implantation of wind turbines as low water depth is required. To that extent, floating wind turbine (FWT) represent a major field of development for the wind industry. Prototypes are currently tested off the coasts of Norway, Japan and Portugal, and a first pilot farm was commissioned at the end of 2017 offshore Scotland.

Fast and accurate simulation methods are crucial for the development of floating wind energy. Simulating floating wind turbine under various aerodynamic and hydrodynamic cases, and especially extreme events, will enable a better dimensioning of both the turbine and the floating structure. This optimization of the geometries currently used is critical for an industrial deployment of this sector, currently noncompetitive compared to more conventional energy sources. Motivated by the development of onshore wind energy, many authors studied the aerodynamic behavior of turbines, e.g. [1], often without representing exactly the blades to reduce the computational effort. The wake effect due to the wind turbine is modeled, which leads to trustworthy aerodynamic results far from the turbine, but also to low precision data near the blades. This limitation can be overcome if full resolution of the flow equation are done in the fluid domain, and some studies consider this, e.g. [2], making them deal with computationally expensive simulations. Hydrodynamic effects also hold a huge influence in these computations, as [3] showed that a combination of high amplitude swell and calm wind can inverse the direction of the relative wind at the rotor. However, dealing with a wind turbine under aerodynamic and hydrodynamic effects requires high efforts in both development and computations. Consequently, only few authors interested themselves in coupled simulations, e.g. [4] or [5]. In a majority of cases, only a component was studied at a time, as did [6], [7] or [8]. The same situation is observed in the literature for experimental studies. Even if a lot of turbines have been studied in wind flumes, few experiments focused on both wind and waves effects. The work in [9] and [13] can be noted, where only particular aerodynamic behaviors have been studied.

The full resolution of the geometries, boundary layers and hydrodynamic effects in the simulation of a floating wind turbine is a real challenge. Events with very different orders of magnitude for both their characteristic time and length are observed in a large computational domain, e.g. between the propagation of the swell and the aerodynamic vortex at the trailing edge of the blades. This problem can be handled using actuator disk methods to approximate the small-scale aerodynamic effects observed at the rotor,

e.g. in [10], or with several overlaid meshes having different levels of refinement, e.g. [4]. This work, funded by the West Atlantic Marine Energy Center (WeAMEC), proposes another methodology based on a unique computational mesh and a monolithic approach. The simulated wind turbine is immersed into the computational mesh, and is defined using a level-set function, while the air/water interface is specified with mixing laws and level-set function. The Navier-Stokes (NS) equations are solved in all the domain using stabilized finite elements (SFE) and a Variational-MultiScale (VMS) formulation. The computational mesh is adapted anisotropically and automatically during the simulation, with its deformations ruled by an error estimator. The generated mesh is well adapted to the simulation of floating wind turbines, as it is refined only at the interfaces and where small-scale events occur.

The methodology used in this work is detailed in Sec. 2, with focus points successively placed on the different steps of the simulation procedure. The first results obtained are presented.

2 Methodology

This Sec. presents the simulation procedure used in ICI-tech. The different software elements are presented in the first four Subsecs. The whole simulation procedure is finally presented in Subsec. 2.5. The validation process then requires the computation of the force. As the meshes used are not body fitted, this computation is not trivial, and the procedure followed is presented in Subsec. 2.6.

2.1 Mesh immersion procedure

The mesh immersion technique consists in a representation of a shape using a single computational mesh. The immersed body can be represented in several ways, the only constraint being the existence of an interior. This characterization enables the construction of a signed-distance function α , presented in Eq. (1) for the immersion of an object ω of frontier Γ in a computational domain Ω .

$$\alpha = \bar{d}(x, \omega) = \begin{cases} d(x, \Gamma) & \text{if } x \in \omega \\ -d(x, \Gamma) & \text{if } x \notin \omega \end{cases}, \quad x \in \Omega. \quad (1)$$

A value for the signed distance is measured at every point of the computational mesh. The reconstruction of the immersed shape can be performed from the sign of this signed distance using a level-set function, e.g. with a Heaviside function returning 1 for the interior ($\alpha > 0$) and 0 for the exterior ($\alpha < 0$). However this Heaviside function introduces discontinuities, which are incompatible with the resolution of the Navier-Stokes equations as presented later on. To that extent a smoothed Heaviside function has been designed, and is presented in Eq. (2).

$$H_\varepsilon(\alpha) = \frac{1}{2} \left(1 + \frac{u_\varepsilon(\alpha)}{\varepsilon} \right), \quad (2)$$

with

$$u_\varepsilon(\alpha) = \varepsilon \tanh\left(\frac{\alpha}{\varepsilon}\right). \quad (3)$$

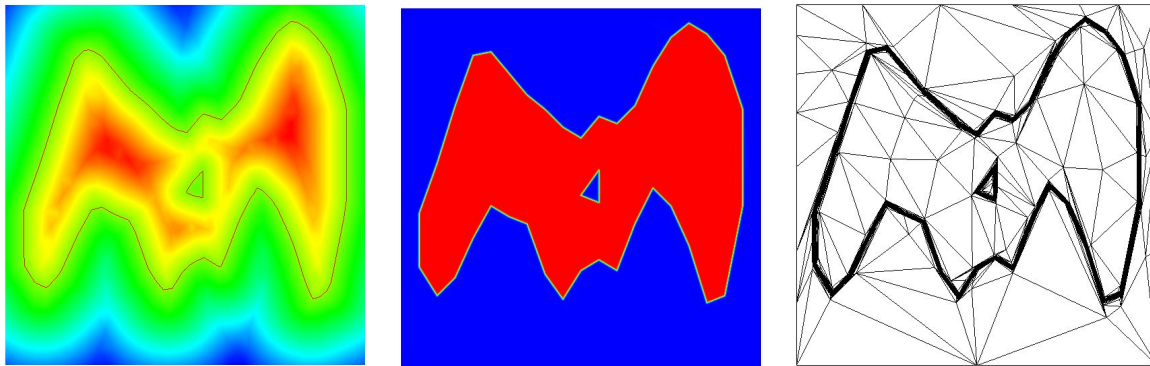


Figure 1: Signed distance α **Figure 2:** Smoothed Heaviside **Figure 3:** Final adapted mesh

This immersion technique introduces a transition area around the frontier of ω , which is of width $2 \times \varepsilon$. The parametrization of H_ε brings flexibility to the immersion procedure. Varying definition of this interface, coupled with an automatic mesh generation positioning computational points in interest zones, guarantees an accurate representation of any geometry immersed. An example of mesh reconstruction is proposed in Figs. 1 to 3. While the signed-distance function presented in Fig. 1 can not achieve mesh immersion by itself, the results obtained with H_ε in Fig. 2 provide an interior, depicted in red. The reconstruction is correctly performed despite the complex shape of ω in this situation. Note that the thin zone at the interface corresponds to the transition area previously mentioned. The mesh obtained at the end of the immersion procedure is presented in Fig. 3. A large majority of computational points are concentrated around Γ , i.e. in the region where the representation of the level-set function is critical to the quality of the reconstruction.

The computational complexity brought by this methodology can become considerable, e.g. when a shape represented by M -elements mesh is immersed in a N -points computational mesh. The signed-distance must be computed at each point of the computational mesh, and if no optimization is performed the distance from the current point to each of the M elements. The complexity obtained is $N \times M$, which can rapidly become unfordable. In the context of floating wind turbines, the computational cost is far too expensive, and optimization has been performed through the construction of a tree structure, named octree.

During the building procedure, the number of elements of the mesh immersed in a computational domain of dimension d is evaluated. If the number found is too important, the computational domain is divided in two along each dimension, thus generating 2^d subdomains. All the elements contained in the initial domain are allocated to the subdomains created, and this procedure is recursively repeated. The construction of the octree ends when the number of elements in the refined subdomains is acceptable or if the maximum depth is reached. When the distance is needed at a given point of the computational mesh, a browsing of the octree enables to select the closest elements from the immersed mesh. The number of distances to evaluate is considerably reduced and the complexity of the immersion procedure is decreased, within a minimal theoretical

complexity of $N \times \log(M)$.

2.2 Anisotropic mesh adaptation

The mesh adaptation procedure is essential for the reconstruction as presented in Subsec. 2.1, but is also a self-standing software unit critical for ICI-tech. The computational mesh is automatically generated from a potentially coarse initial mesh and tends to gather a maximum of points in the interest zones. The theory proposed by [11] is based on a principle of equidistribution of the error in all the computational domain. An a posteriori error estimator is built at each node of the computational mesh. The interest zones previously mentioned are defined from this estimator: high error areas need to be refined and vice versa. The evolution of the error is examined along the edges of the mesh, and important variations can lead to points insertion. Metrics are built at each node from edge errors in order to deform the mesh, always with an error minimization goal. These metrics feature scaling factors for each dimension as refinement is conducted only along the representative dimensions. The mesh generated is anisotropic, which enables to reduce largely the amount of points needed in the simulations. This procedure is repeated iteratively, until an appropriated mesh is obtained.

The adaptation depends on the construction of the error estimator. In the context of mesh immersion, as proposed in Subsec. 2.1, the error estimator is built from the results provided by the level-set function. As a smoothed Heaviside function is used, important gradients are found around the interfaces of the objects immersed, while the function is quasi-constant elsewhere. The level-set function printed in Fig. 2 encounters high gradients in the green region around the interface, which explains the high concentration of points in these areas found in the mesh presented in Fig. 3. On the contrary large cells are generated in zones where colors remain constant, e.g. at the bottom of the computational domain. Whenever several bodies are immersed in the same computational mesh, the immersion procedure is conducted for each body, i.e. a level-set function is defined for each immersed element. In order to guarantee correct adaptation for each body, the construction of the error estimator takes every level-set function results into account. Some immersed bodies can be weighted in the error estimator in order to prioritize their adaptation.

However level-set meshes as depicted in Fig. 3 are not suitable for the resolution of fluid flows, because large cells introduce numerical dissipation and thus miss sub-scale flows. The mesh adaptation needs to take into account the fluid effects occurring, and consequently the error estimator is derived to take into account the velocity output. The error estimator is composed from both the phases of the immersed bodies and the velocity. The mesh obtained with a $Re = 1000$ -flow, horizontally oriented towards the right of the computational domain, past a cylinder is presented in Fig. 4. At the beginning of the simulation the major part of cells are concentrated around the circle. After some increments high concentration of cells are observed in the wake as the adaptation procedure tracks the vortices. Note that using the phases for the adaptation is not necessary for obtaining accurate fluid flows. However the full CFD paradigm proposed in Subsec. 2.4 requires the capture of boundary layers, and to that extent the adaptation on the phases

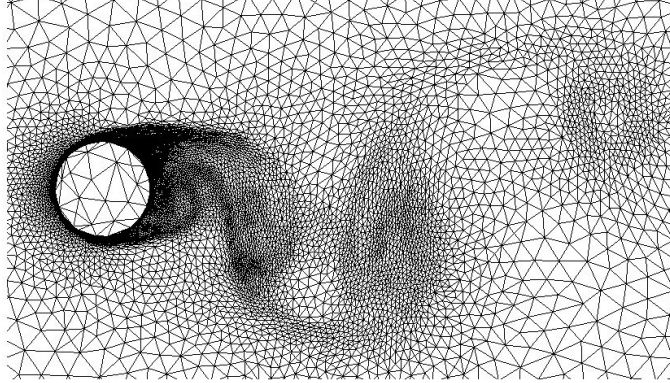


Figure 4: Mesh for a flow past a cylinder at $Re = 1000$

of the immersed body is an advantage of the present method.

2.3 Mixing laws

Once the mesh immersion procedure ends, the resolution of fluid flows can be solved. It is done in ICI-tech with an incompressible resolution of the NS equations using the VMS formulation. But for the NS problem to be solved, the characteristics of the flow at each point of the computational mesh have to be known. To that extent mixing laws are applied with use of the level-set function defined in Eq. (2). Viscosity and density are specified for each of the phase in the computation, and mixing laws are applied. A linear law is presented in Eq. (4) for the immersion of an object of viscosity η_i in a domain of viscosity η . At a point whose distance to the immersed element is α , the viscosity η_p obtained is presented. If several elements are immersed, the mixing laws are applied as many times as needed. The linear mixing law presented is appropriated when the different viscosities found in the simulation have similar orders of magnitude. However when large disparities are encountered, other laws may be more appropriated, e.g. logarithmic ones.

$$\eta_p(\alpha) = \eta_i H_\varepsilon(\alpha) + \eta(1 - H_\varepsilon(\alpha)). \quad (4)$$

The same mixing laws are applied for density, and consequently every point of the computational domain possesses its own viscosity and density. To make a link with the example proposed in Subsec. 2.1, the viscosity of the object depicted in red is η_i , while the part of the domain in blue has a viscosity of η . The transition region, in green, has a viscosity varying between η and η_i depending on the signed-distance function. The presence of an important number of points in this transition area restricts the difference of viscosity found in a cell, which is helpful in terms of NS resolution.

2.4 Resolution of the Navier-Stokes equations

ICI-tech proposed a monolithic approach based on stabilized finite elements where the incompressible NS problem, presented in Eq. (5) with \mathbf{v} velocity and p pressure, is solved

with the VMS formulation.

$$\begin{cases} \forall(\mathbf{w}, q) \in V_0 \times Q, \\ \rho(\partial_t \mathbf{v}, \mathbf{w}) + \rho(\mathbf{v} \cdot \nabla \mathbf{v}, \mathbf{w}) + 2\nu \boldsymbol{\varepsilon} \mathbf{v} : \boldsymbol{\varepsilon} \mathbf{w} - (p, \nabla \cdot \mathbf{w}) = (\mathbf{f}, \mathbf{w}) \\ (\nabla \cdot \mathbf{v}, q) = 0 \end{cases} \quad (5)$$

The VMS paradigm operates as an implicit-LES (Large Eddy Simulation). Velocities are split between the coarse scales, superior to the local mesh size which are solved, and sub-scale, which are modeled. This is expressed in Eq. (6), with \mathbf{v}_h coarse scale velocity and \mathbf{v}' subgrid-scale velocity. In LES a viscous term are incorporated in the NS equations to compensate for the effect of subgrid-scale turbulence. The VMS reformulates the NS problem to express the subgrid-scale velocities, and to mathematically evaluate their influences. Presented in Eq. (7), the VMS formulation of the NS problem features stabilization terms (τ_K , τ_C) and residuals (\mathcal{R}_M , \mathcal{R}_C) coming from this evaluation of subgrid-scales.

$$\mathbf{v} = \mathbf{v}_h + \mathbf{v}' \quad (6)$$

$$\begin{cases} \forall(\mathbf{w}_h, q_h) \in V_{0,h} \times Q_h, \\ \left\{ \begin{aligned} &\rho(\partial_t \mathbf{v}_h, \mathbf{w}_h) + (\rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h) - \sum_{K \in \mathcal{T}_h} (\tau_K \mathcal{R}_M, \rho \mathbf{v}_h \nabla \mathbf{w}_h)_K \\ &+ 2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h) - (p_h, \nabla \cdot \mathbf{w}_h) + \sum_{K \in \mathcal{T}_h} (\tau_C \mathcal{R}_C, \nabla \cdot \mathbf{w}_h)_K = (\mathbf{f}, \mathbf{w}_h) \\ &(\nabla \cdot \mathbf{v}_h, q_h) - \sum_{K \in \mathcal{T}_h} (\tau_K \mathcal{R}_M, \nabla q_h)_K = 0 \end{aligned} \right. \end{cases} \quad (7)$$

More details on stabilization terms and residuals can be found in [11]. Nonetheless it can be noted that the stabilization terms depend on both $\|\mathbf{v}_h\|$ and the characteristic length of the mesh cell currently evaluated.

2.5 Simulation procedure

The simulations with ICI-tech rely on mesh adaptation, phase reconstruction and resolution of the NS equations, detailed in the previous Subsec. The simulation procedure, presented in Fig. 5, starts with a mesh adapted on the level-set obtained after immersion iterations. Mesh immersion is performed with this mesh, and mixing laws are applied. The NS problem is then solved with a given time step, providing velocity and pressure at every point of the computational domain. The mesh is adapted as proposed in Subsec. 2.2, based on weighted velocities and level-set results. Consequently the mesh newly generated will concentrate cells where fluid effects occur. This procedure is repeated until full time simulation is performed.

Even if anisotropic meshing enables a significant reduction in the number of points needed in the simulations, the full-CFD paradigm proposed by ICI-tech remains computationally expensive. In order to make ICI-tech competitive, optimization of computational costs have been conducted in two different ways. The first orientation has been the

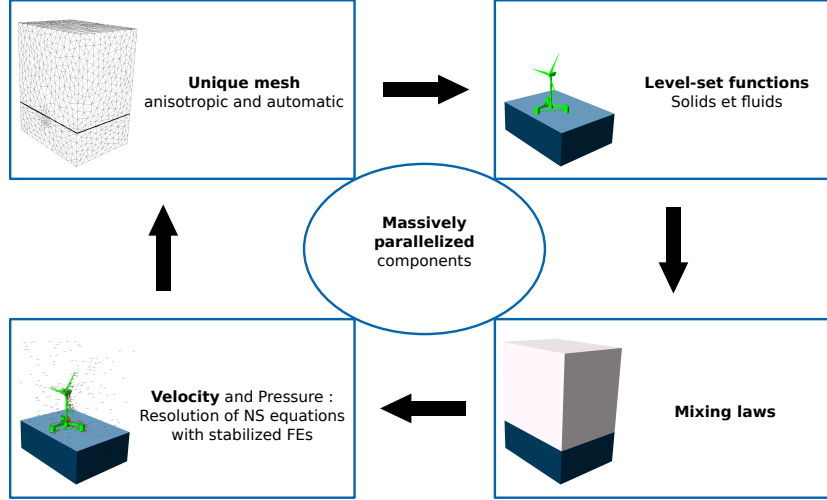
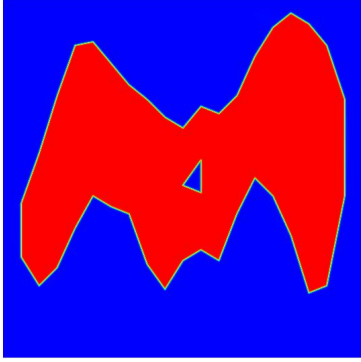
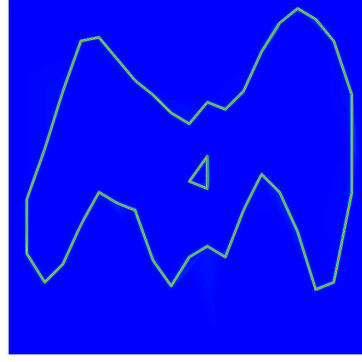


Figure 5: Simulation procedure with ICI-tech

limitation of the complexity of the algorithms, e.g. through the octree implementation briefly discussed in Subsec. 2.1. The code has also been massively parallelized in order to enable the use of a large number of cores. Particular attention has been placed on partitions during parallel meshing and NS resolution, details can be found in [12].

2.6 Computing the force applied on an immersed element

The computation of the force applied on a object Ω is not trivial in the context of mesh immersion, as the mesh is not body fitted. However the concentration of point around the interfaces, highlighted in Fig. 3, enable accurate evaluation of the loads. The forces are computed a posteriori from the results of velocity \mathbf{v} and pressure p obtained after the NS resolution. A reconstructed velocity gradient is built and the stress tensor σ is computed at every point of the mesh, with η viscosity and \mathbb{I} identity matrix. The stress tensor accounts for all the fluid effects occurring, and to evaluate the loads applied on Ω , the computation of local normal constraint \mathbf{T}_{local} is required and performed in Eq. (8). The definition of a vector normal to Ω is necessary at that point, but as the computational points are not disposed along $\partial\Omega$, this determination is not immediate. It has been chosen to define \mathbf{n} as a normalized gradient of the phase ϕ obtained in the immersion of Ω . This choice brings different possibilities. First of all it enables to compute \mathbf{n} on every point of the mesh using an existing data. The gradient of the phase is computed and no projection are performed, which brings computational savings. This definition also introduces interesting behavior when $\nabla\phi$ is computed outside the transition zone observed during immersion. As ϕ becomes rapidly quasi-constant after this area, the norm of the gradient tends towards zero. This sets a natural filter which selects the points located nearby $\partial\Omega$, which have the greatest impact on the force generation. The local normal constraints are then computed at each point of the computational mesh from the corresponding stress tensor and normal vector. The force F applied on Ω is then deduced from an integration of the local normal constraints over $\partial\Omega$, as in Eq. (9).


Figure 6: Smoothed Heaviside

Figure 7: Dirac function δ_ε

$$\mathbf{T}_{local} = \boldsymbol{\sigma} \cdot \mathbf{n}, \text{ with } \begin{cases} \boldsymbol{\sigma} = \eta(\nabla \mathbf{v} + \nabla \mathbf{v}^T) - p\mathbb{I} \\ \mathbf{n} = -\frac{1}{\|\nabla \phi\|} \nabla \phi \end{cases} \quad (8)$$

$$\mathbf{F} = \int_{\partial\Omega} \mathbf{T}_{local} dS \quad (9)$$

As the mesh is not body-fitted, the approach proposed by [14] is used. The expression is reformulated to turn the integral over $\partial\Omega$, condition that points can hardly meet, into an integral over Ω . The force computed over the transition zone F_ε is presented in Eq. 10.

$$\mathbf{F}_\varepsilon = \int_{\Omega} \delta_\varepsilon \mathbf{T}_{local} dV, \text{ with } \delta_\varepsilon = \frac{1 - \phi^2}{2\varepsilon} \quad (10)$$

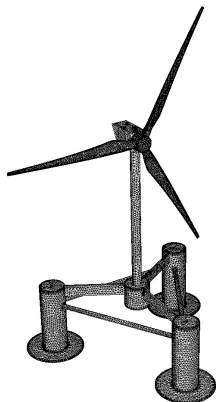
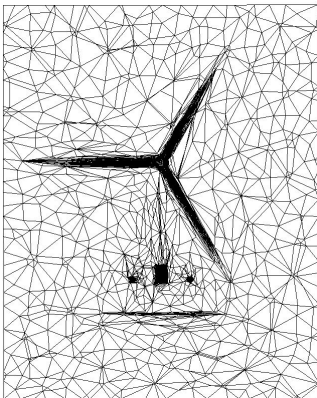
This formulation is inherited from the level-set definition. The ε parameter, characterizing the width of the transition area, is defining the span of the integration zone. Moreover the dirac function δ_ε introduced is directly derived from ϕ . The level-set function and the dirac one are respectively depicted in Figs. 6 and 7. The closer a point is located from the interface, the higher value is obtained with the dirac function. Consequently the points located far from the interface have a quasi null influence on \mathbf{F}_ε , and can be neglected. The convergence of \mathbf{F}_ε when the width of the transition area tends towards zero has been discussed by [14], who proved that the limit obtained is \mathbf{F} .

$$\lim_{\varepsilon \rightarrow 0} \mathbf{F}_\varepsilon = \mathbf{F} \quad (11)$$

The precision of the force representation is consequently directly depending from the width parameter, i.e. the quality of the force obtained is determined by the immersion process.

3 Application to floating wind turbines

This Sec. focuses on the works realized for the simulations of floating wind turbines. The results obtained with the mesh adaptation procedure are presented in Subsec. 3.1.

**Figure 8:** Immersed mesh**Figure 9:** Slice of adapted mesh**Figure 10:** Reconstructed wind turbine

The validation of the solver is currently being performed, and Subsecs. 3.2 and 3.3 detail the procedures followed for validation of respectively monophasic flows and wave generation.

3.1 Floating wind turbines reconstruction

The reconstruction of floating wind turbines have been studied in order to evaluate the number of nodes required for an accurate representation of the geometries. The surface mesh of a prototype studied in wind flume by [13], composed of approximately $75K$ facets and depicted in Fig. 8, is immersed in a computational domain containing about $120K$ points. A slice of the mesh obtained at the end of the immersion procedure is presented in Fig. 9. This slice intersects the rotor, and consequently the high concentration of points afford to guess the position of the blades. The width parameter of the level-set, named ε in Subsec. 2.1, is of about $1/1000$ -th of the overall wind turbine size. The order of precision of the immersion is of the decimeter for a full-scale wind turbine. This precision might seem very low, particularly at the tip of blades, but the reconstructed wind turbine depicted in Fig. 10 is appropriated for visualization purposes. The slice of the mesh highlights the presence of coarse cells far from the turbines. This mesh, which is too coarse for flow simulations, is used to initiate the computations. The adaptation procedure coupled to the resolution of the Navier-Stokes problem generates refinement based on the velocity of the flow, which will lead to thinner cells in interest zones. Note that more than one billion points are required to obtain the same precision with a regular grid.

This case gives an occasion to highlight the computational savings provided by the octree. The construction of the octree was performed as presented in Subsec. 2.1 with a limit of 200 elements per subdomain and a max depth of 12. The cost of the distances computation for the immersion of the wind turbine mesh (Fig. 8) in the adapted computational mesh (Fig. 9) is presented in Tab. 1. The focus has been placed on the computation of distances because this step features the highest complexity. Moreover the evaluation of

distances is often required during a simulation cycle, as explained in Subsec. 2.5, and its optimization was a need. The computational time required for distance evaluations was divided by about 235 only by the addition of the octree. The browsing of the octree to select the closest neighbors for each point enabled to largely reduce the number of distances to evaluate, as only 0.081% of the computations were processed. The cost of the octree browsing increases the average time needed to compute a distance by a factor of about 5, which is reasonable considering the huge limitation in the number of evaluations done. This example assessed the huge interest of the octree structure for distance computation algorithm, and by extension for mesh immersion procedure.

	Distances evaluated	Time for distance evaluation	Time per distance (s)
No Octree	$\sim 8693M$ (100%)	9min, 51s (100%)	$6.8 \cdot 10^{-8}$
Octree	$\sim 7M$ (0.081%)	2.5s (0.4%)	$3.5 \cdot 10^{-7}$

Table 1: Influence of octree for wind turbine immersion

The octree implementation intended to limit the complexity of the distance algorithm. The scalability of the immersion procedure has then been studied with a test case of 100 wind turbines. The wind turbine mesh presented in Fig. 8 was duplicated and randomly dispersed in a computational domain of size $50 \times 50 \times 2.5$. As a matter of comparison the dimensions of the computational domain used for the reconstruction of a single wind turbine were $2 \times 2 \times 2.5$. The immersed mesh consequently features 100 times more facets, the results of immersion are presented in Figs. 11 and 12. The computational mesh used for the immersion contains $18M$ nodes for the same precision as in the single turbine test case. The number of nodes used has been multiplied by about 150, while the immersed mesh was 100 times bigger. This can be partly explained by the increase in computational mesh size, and even with this explanation the rise in the number of nodes needed has been well limited. Moreover the increase in computational times can be controlled with a corresponding increase in the number of processors used. The disposition of the floating wind turbines does not reproduce a wind farm, however the computational cost required for the immersion has the same order of magnitude, which proves the potential of ICI-tech for offshore wind applications. Note that the colors depicted on the wind turbines correspond to the partitions used for the immersion. They are well divided, as the processors communicate between each other to share the computational load.

3.2 First validation towards high Reynolds number single-phase flows

ICI-tech is currently used for applications such as injection in the context of composites. The flows studied commonly have low Reynolds number. On the contrary at the tip of blades the combination of wind speed and blade rotation speed generate flows with Reynolds of several millions. The validation of the flow solver is currently conducted in order to validate the results obtained for those Reynolds. To that extent drag and lift coefficients have to be calculated, which required the computation of the force applied on the immersed shape. The evaluation of the force is presented in Subsec. 2.6. Once the value of the force is obtained, the drag and lift coefficient C_p and C_l can be calculated, from

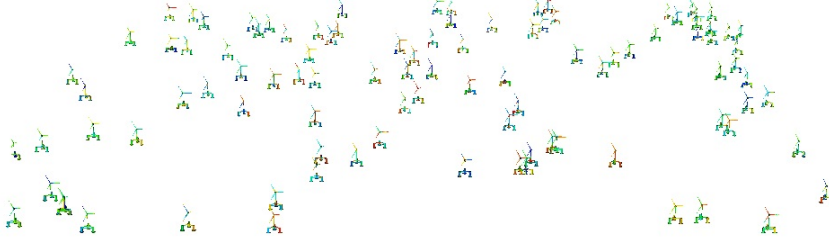


Figure 11: Reconstruction of 100 floating wind turbines

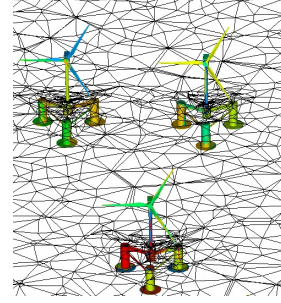


Figure 12: Zoom on 3 WTs and mesh slice

Eqs. 12 and 13 with A cross sectional area. The pressure coefficients are evaluated with Eq. 14 with p_∞ , ρ_∞ and V_∞ respectively pressure, density and velocity in the freestream.

$$C_d = \frac{2}{\rho U^2 A} \mathbf{F} \cdot \mathbf{x} \quad (12)$$

$$C_l = \frac{2}{\rho U^2 A} \mathbf{F} \cdot \mathbf{y} \quad (13)$$

$$C_p = \frac{p - p_\infty}{\frac{1}{2} \rho_\infty V_\infty^2} \quad (14)$$

The validation started on 2D test cases, where flow passing NACA profiles were studied. The results obtained with ICI-tech (CFD, SFE, VMS) were compared against data from [15], where a Diffused Vortex Hydrodynamics method is used. A particular focus was placed on the test cases that featured NACA 0008 profiles, as wind turbine blades are commonly composed of both NACA and DU profiles. Validation is currently being performed on test cases whose Reynolds number are $Re = 2000$ and $Re = 6000$, and the influence of all the different parameters used in computations are highlighted. The width parameter ϵ and the number of points composing the computational mesh are particularly overlooked.

The flow passing a NACA-0008 with an orientation of 4° immersed in a constant uniform flow of $Re = 2000$ is studied, to draw a comparison with the results of [15]. Different values of ϵ are studied in order to evaluate the influence of this parameter, while the number of nodes in the computational meshes was kept constant at $80k$. This number of nodes is far too important for this kind of simulation, which guaranteed that it would not be a blocking factor in the simulations. The vorticity fields obtained with both codes are presented in Fig. 13, and the mesh generated by ICI-tech and used in the computation for $\epsilon = 2 \cdot 10^{-4}$ is depicted in Fig. 14. An important numerical diffusion is highlighted on the results from ICI-tech, which could be limited by works on the quality of the mesh adaptation. Figs. 15 and 16 present the results obtained with a $Re = 2000$ -flow around a NACA-0008 profile, for different ϵ (L_{max}). Those results are compared against DVH data provided by [15]. At steady state, interesting precision is met for drag, while errors remain for lift or pressure coefficients. The dependence of the results towards the width

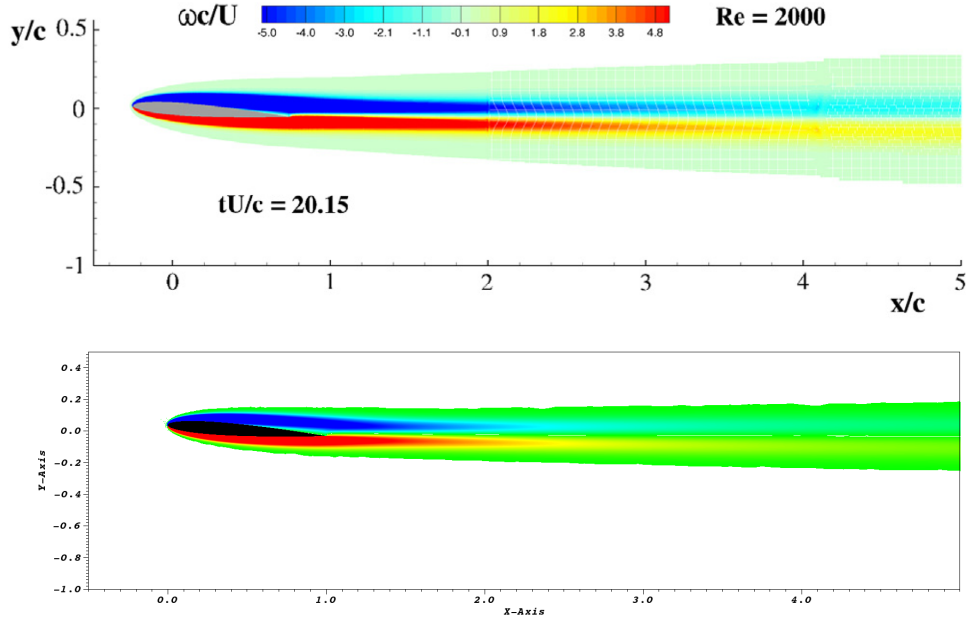


Figure 13: Up: Vorticity field around NACA-0008 at $Re = 2000$, from [15]. Down: Results obtained with ICI-tech.

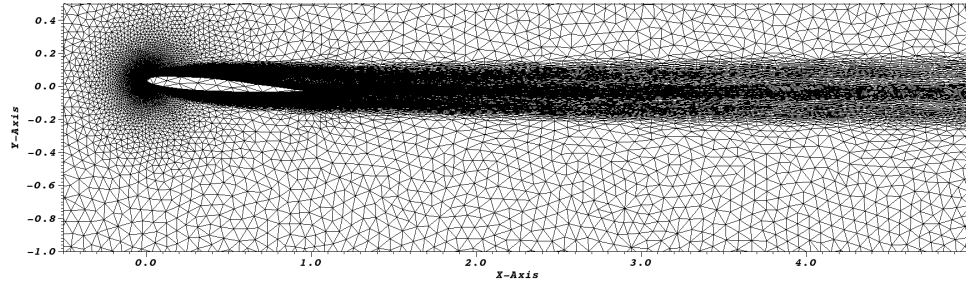


Figure 14: Mesh obtained for the NACA-0008 $Re = 2000$ test case with ICI-tech.

parameter seems to be limited. The noise observed on the C_p curves seems to be due to the anisotropic meshing of the elements immersed. Further researches are currently conducted in order to improve the results presented.

Those $Re = O(1000)$ computations can not be considered as high Reynolds flows. The validation will then continue on higher Re , e.g. with data from [16]. 3D NACA profiles will then be studied, before performing simulations on blades.

3.3 Wave generation and free-surface tracking

Accurate resolution of the free-surface dynamics are crucial for a good resolution of the movements of floating wind turbines. The simulations presented in the next paragraphs aim to track precisely the free-surface. The mesh will need to be adapted to follow the air/water interface, which is handled in ICI-tech by means of the convected level-set method introduced in [17]. This technique relies on the introduction of transport terms into level-set function, which is modified to introduce an advection velocity. The

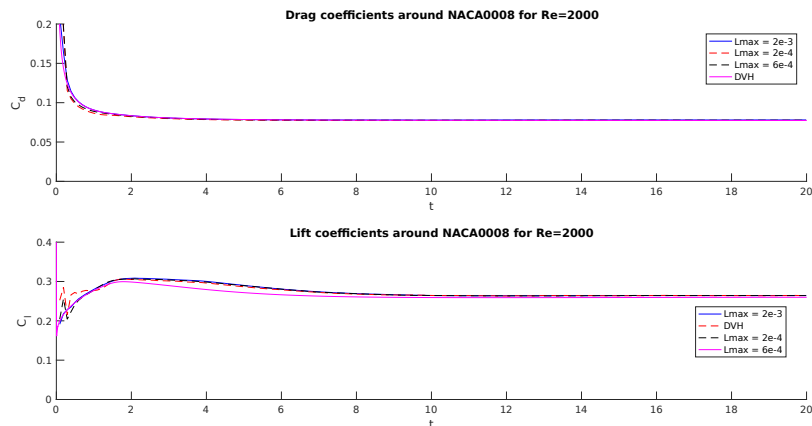


Figure 15: Drag and lift coefficients for flow around a NACA-0008 profile at $Re = 2000$ for different ε

free-surface movements being taken into account, the level-set function will consequently better capture the interface.

A numerical wave-maker has been implemented. A piston-type wave maker was chosen, whose movements are determined using HOS-NWT [18]. From a specified wave field, HOS-NWT returns the movements of the wave generator. Consequently the validation will consist on a tracking of the free-surface generated, and its comparison with the wave field used as HOS-NWT input.

A wave field of period 0.55s and amplitude 0.1m propagated from the left of the computational domain have been generated in ICI-tech. The preliminary results presented hereafter intend to reproduce an air/water interface, but have been realized with viscosities a hundred-times bigger for each phase. This artefact enabled to better stabilize the computations, but will have to be suppressed in order to get high precision results. A ε of $6 \cdot 10^{-3}$ and $40k$ points are used for the computational mesh. As for the single-phase computations, this important amount of nodes enable to study the influence of the width parameter only.

The free-surface obtained with ICI-tech is drawn in blue in Fig. 17, while the wave field input in HOS-NWT is presented in red. If the fidelity of the first wave generated is interesting, the ICI-tech free-surface results progressively feature more damping and a very important noise at the right end of the domain. The damping can be partially explained by the higher viscosities chosen for the fluids, but the influence of numerical diffusion will have to be overlooked. The noise observed will be removed by the addition of a numerical absorption area at the right end of the domain. The mesh generated and used by ICI-tech is presented on Fig. 18. The majority of the cells are concentrated along the free-surface, and a high concentration is also observed at the right end of the basin. The numerical absorption area will need to limit the refinement in this area, in order to limit the number of nodes used in that region and to increase the numerical dissipation.

Further researches are being performed on the 2D basin, before a transition towards a 3D numerical wave tank. The generation of complex wave field will finally be realized.

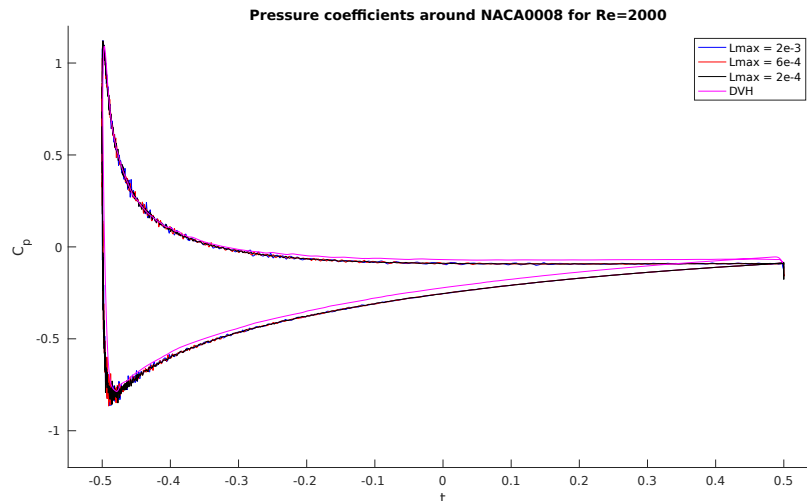


Figure 16: Pressure coefficients for flow around a NACA-0008 profile at $Re = 2000$ for different ε

4 Conclusion and outlook

ICI-tech seems to offer potential for accurate simulation of floating wind turbines, even if lot of development are still needed. The uniqueness of the computational mesh coupled to the immersion procedure provides a lot of flexibility to the simulation, as an adapted mesh can be generated iteratively from a coarse initial. The code uses automatic meshing routines that enable the reduction of the size of the computational meshes. The cost commonly required for full-scale LES simulation can be reduced, while the same precision on the results is obtained, thanks to the VMS solver implemented. The code is undergoing validation, both on aerodynamics and hydrodynamics. Single-phase flows will be validated in 2D with NACA profiles, before a transition to 3D. 3D NACA profiles will be studied, and finally flows passing blades and rotor will be overlooked. Free-surface simulations will focus on 2D, with regular and irregular waves. The same steps will be repeated in 3D, before studying the interactions between wave fields and floating structures.

5 Acknowledgments

This work is funded by the West Atlantic Marine Energy Center. This work was performed by using HPC resources of the Centrale Nantes Supercomputing Centre on the cluster Liger and supported by a grant from the Institut de Calcul Intensif (ICI) under the project ID E1611150 / 2016.

REFERENCES

- [1] Hansen, M. O. L. and Srensen, J. N. and Voutsinas, S. and Srensen, N. and Madsen, H. A. State of the art in wind turbine aerodynamics and aeroelasticity. *Progress in aerospace sciences*, vol. 42, pp. 285–330, 2006.

- [2] Bazilevs, Y. and Korobenko, A. and Deng, X. and Yan, J. Novel structural modeling and mesh moving techniques for advanced fluid–structure interaction simulation of wind turbines. *IJNME*, vol. 102, pp. 766–783, 2015.
- [3] Sebastian, T. and Lackner, M. Offshore floating wind turbines - An aerodynamic perspective. *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, vol. 720, 2011.
- [4] Quallen, S. and Xing, T. CFD simulation of a floating offshore wind turbine system using a variable-speed generator-torque controller. *Renewable Energy*, vol. 97, pp. 230–242, 2016.
- [5] Leble, V. and Barakos, G. Demonstration of a coupled floating offshore wind turbine analysis with high-fidelity methods. *Journal of Fluids and Structures*, vol. 62, pp. 272–293, 2016.
- [6] Wu, C.-H. K. and Nguyen, V.-T. Aerodynamic simulations of offshore floating wind turbine in platform-induced pitching motion. *Wind Energy*, 2016.
- [7] Tran, T.-T. and Kim, D.-H. The platform pitching motion of floating offshore wind turbine: A preliminary unsteady aerodynamic analysis. *JWEIA*, vol. 142, pp. 65–81, 2015.
- [8] Yan, J. and Korobenko, A. and Deng, X. and Bazilevs, Y. Computational free-surface fluid–structure interaction with application to floating offshore wind turbines. *Computers & Fluids*, 2016.
- [9] Courbois, A. Étude expérimentale du comportement dynamique d’une éolienne offshore flottante soumise l’action conjuguée de la houle et du vent. Ecole Centrale de Nantes, 2013.
- [10] Sanderse, B. and Pijl, S. and Koren, B. Review of computational fluid dynamics for wind turbine wake aerodynamics. *Wind energy*, vol. 14, pp. 799–819, 2011.
- [11] Coupez, T. and Hachem, E. Solution of high-Re. incompressible flow with stabilized finite element and adaptive anisotropic meshing. *CMAME*, Vol. 267, pp. 65–85, 2013.
- [12] Digonnet, H. and Coupez, T. and Laure, P. and Silva, L. Massively parallel anisotropic mesh adaptation. *IJHPCA*, 1094342017693906, 2017.
- [13] Lacaze, J.-B. and Molin, B. and Remy, F. and Branger, H. and Luneau, C. Small scale tests of floating wind turbines in the wind and wave flume of Luminy. *14th journées d’hydrodynamique*, Val de Reuil, France, 18-20 Nov 2014.
- [14] Brackbill, J. and Kothe, D. and Zemach, C. A continuum method for modeling surface tension. *Journal of Computational Physics*, vol. 100, pp. 335–354, 1992.

- [15] Rossi, E. and Colagrossi, A. and Durante, D. and Graziani, G. Simulating 2D viscous flow around geometries with vertices through the Diffused Vortex Hydrodynamics method. *CMAME*, vol. 302, pp. 147–169, 2016.
- [16] Bak, C. and Fuglsang, P. and Johansen, J. and Antoniou, I. Wind Tunnel Tests of the NACA 63-415 and a Modified NACA63-415 Airfoil. Forsøgsanlæg Risø, 2002.
- [17] Ville, L. and Silva, L. and Coupez, T. Convected level set method for the numerical simulation of fluid buckling. *IJNMF*, vol. 66, pp. 324–344, 2011.
- [18] Ducrozet, G. and Bonnefoy, F. and Le Touzé, D. and Ferrant, P. A modified high-order spectral method for wavemaker modeling in a numerical wave tank. *EJMBF*, Vol. 34, pp. 19–34, 2012.

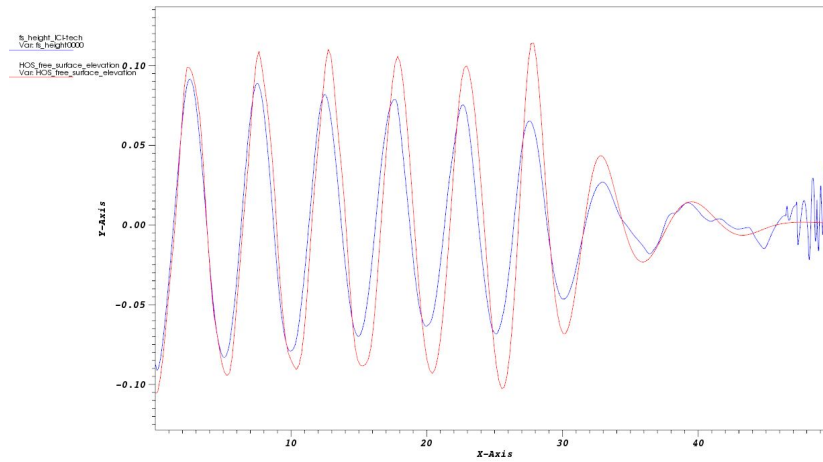


Figure 17: Comparison between a wave field generated using ICI-tech and the results from HOS-NWT

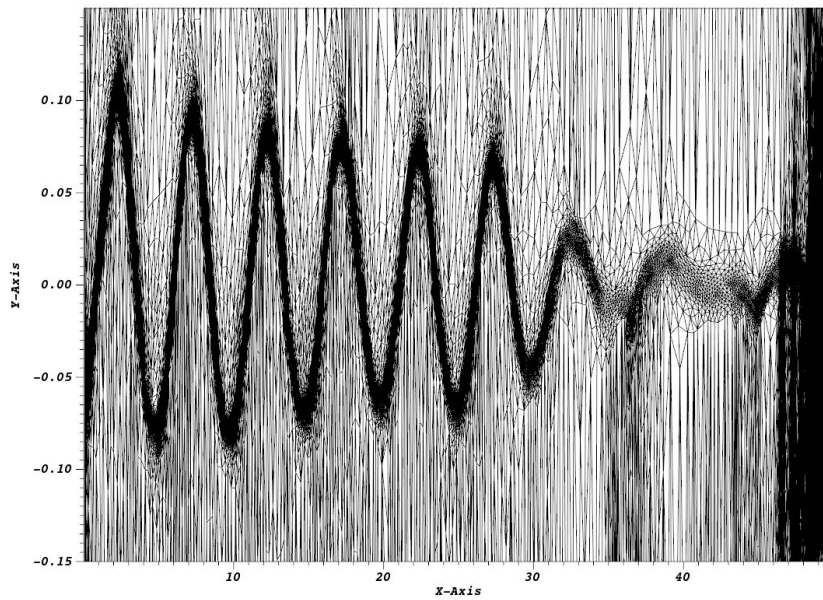


Figure 18: Mesh generated by ICI-tech for the wave maker test case