### Slide 01

It is my pleasure to be here today and share the recent developments I have made in the numerical framework for simulating turbulent flow past wind turbines operating in Turbulent Inflow.

### Slide 02

The outline of my presentation is as follows. First, I will present the motivation behind studying wind turbines as a vital source of renewable energy and outline the main objective of the first year of this three-year research project.

Next, I will provide an updated overview of the numerical framework by describing the different properties and optimizations implemented in the solver in the last six months.

Finally, I will present some results and discuss future developments in the coming six months.

### Slide 03

Morocco, like many other countries, is facing a major challenge: ensuring energy stability and keeping the light on while addressing the urgent task of combating climate change.

Wind energy emerges as a crucial ressources, with some countries achieving penetration levels in power generation of over 30%.

To extend these benefits globally, one of the main factor that should be reduced is the levelized cost of energy (LCOE) to make wind sources more competitive with traditional fossil fuel-based sources of electricity.

As far as Morocco is concerned, a challenging plan was initiated a decade ago to produce over 52% of its electricity through renewable sources by 2030.

This initiative is underpinned by Morocco's geographic location and favorable climate conditions, which provide a strong foundation for harnessing wind energy in particular.

As a matter of fact, the wind generation potential is measured at approximately 5000 TWh per year, and a potential useful capacity, which measures the maximum amount of electricity that can be consistently produced and supplied to meet the demand, of 25,000 MW.

Now, in response to the growing demand, wind turbines are increasing in size with rotor diameters exceeding 200 meters.

We are talking about colossal turbines whose overall perforamnce is significantly depending on different atmospheric conditions.

Therefore, it's essential to study how airflow behaves around wind turbines with flexible blades, especially considering complex factors like aero-servo-elasticity coupling.

Among the various numerical approaches to achieve this goal, Large Eddy Simulation (LES) stands out as a powerful tool.

It provides a high-fidelity representation of the unsteady flow field at the wind farm scale and serves as an affordable tool to calibrate wind farm layout optimization models and engineering numerical models of complex phenomena, such as dynamic wake meandering, wake steering, blade loading cycles (including wake turbulence), and the blockage effect due to the turbine row.

### Slide 04

The use of this approach is still quite challenging due to the wide range of length and time scales that are required to be resolved as the Reynolds numbers involved are so high.

In light of these limitations, an efficient massively parallel implementation becomes necessary to make it computationally achievable.

That's why, in this first-year project, the goal is to develop a massively parallel framework that combines Large Eddy Simulation (LES) and the Actuator Line Modeling (ALM) technique. The focus is on achieving optimal workload balancing of the numerical simulator and minimizing return time using state-of-the-art methods.

### Slide 05

The numerical solver I have been developping in the last nine month is an unstrucutred finite volume solver designed for low-Mach number Large-Eddy Simulation.

The guiding principle of the solver is the highly parallel aspect capacities as attested by the IO handling that can support to few billions of cells.

The resolution of the NS equations is based on classical projection methods, in which the critical step of solving the Poisson equation is handled through highly efficient and stable Deflated Preconditioned Conjugate Gradient in combination with a residual recycling strategy and and an efficient adaptation of the convergence criteria to reduce the number of iteration to convergence.

Several turbulence models are now integrated into the numerical solver, including the Constant Smagorinsky, Localized Dynamic Smagorinsky, Vreman, WALE, and the $\sigma$-model.

Noteworthy to mention that the implementation of these models, especially the Dynamic Smagorinsky variants, is more challenging for unstructured grids compared to structured ones.

The solver now can be used efficiently on thousands CPU cores thanks to its two-layer parallelism technique that allows to avoid the issues of the classical single domain decomposition of the mesh and optimize accesses to memory.

**### Slide 06**

The standard Single Domain Decomposition consists of a simple splitting of the domain into a number of subdomains equal to the number of processors used for the computation, which is performed thanks to the Metis and Scotch Library.

The variables on the subdomain are stored in the local memory, and MPI communication protocol is used for the exchange of values at the frontiers between subdomains.

Even though this technique gives interesting results, but it holds several drawbacks for massively parallel solving, especially regarding the issues of accesing data in the cache, load balancing and local mesh refinement.

Moreover, if there are too many cells for each processor as it is generally the case for realistic wind turbine simulations, the computation is slowed down by cache memory overlappings.

**### Slide 07**

To avoid this issues, an additional level of subdomains is introduced, splitting the subdomain assigned to a given processor into groups, which are small enough to fit into L3 or even L2 cache.

The inter-element groupe communication are either local and based on openMP intra-processure request.

This approach not only mitigates the challenges associated with load balancing and cache memory overlaps but also enhances the capabilities of the DPCG algorithm, making it faster.

**### Slide 08**

As I mentionned, one of the main advantages of the double domain decomposition is to enhance dynamic load blanacing and dynamic mesh refiment.

The solver now incroporate a fully parallel dynamic mesh adaptation technique to improve the resolution in physically-relevant zones to mitigate the computational cost that works for the moment only in the 2D simulation.

This dynamic mesh adaptation is based on the MMG3D libary.

This library is based on local mesh modifications that is conducted at the element groupe level and allows to reach some very interesting precision for a given cell count as illustrated in this slide, where the adaptation is based on the local vorticity.

### Slide 09

The flow here is a simple flow around a cylinder at Reynolds number of 500. All the main carateristics that I have described can be show here. Typically, the double domain decomposition, and the dynamic mesh refinement.

### Slide 10

Without going into the mathematical formalism of the ALM method, I would like to emphasize that the method is fully implemented in the solver, incorporating various optimizations and corrections.

It is important to note that the primary challenge arises from the fact that all the steps are conducted on a non-cartesian grid, making the implementations considerably more complex when compared to structured grids, as is the case in most open-source solvers.

Furthermore, in light of several improvements made to the solver, various corrections have been implemented to account for 3D stall delay, tip/hub losses, dynamic stall, and filtered lifting line.

### Slide 11

As the implementation is designed with the spirit of a massively parallel framework, the final step of the ALM, involving the mollification process, is applied to a small and reduced number of cells in the mesh to reduce CPU costs.

In the case of a structured mesh, this process is straightforward as all the nodes are easily identified. However, in the case of an unstructured grid, I have developed a novel three-step methodology to optimally identify the nodes where the optimization is applied, considering the double domain decomposition.

In terms of MPI exchange communication, currently, each turbine is recognized by every processor involved in the computation, leading to duplication of information, which results in a relatively high costs. However, this cost increase remains around two percent even with the addition of two turbines in the computational domain. Nevertheless, I need to refine this aspect further to optimize the process.

### Slide 12

Another optimization that is implemented now in the solver concerns the modification of the limiting time step when using the actuator line method.

Similar to the classical definition of CFL, a corresponding number can be introduced to quantify the actuator element displacement based on the local cell size during the time step, given by CFL_rotor.

If CFL_rotor >> 1, this will introduce discontinuities in the projected forces of the trailing timesteps and therewith introduces purely numerical fluctuations in the velocity field.

To prevent this numerical fluctuations, a time mollification mollify the forces with a substeping process.

This methodology has been proved to reduce the computational time without reducing the fidelity of the simulation while keeping the code stable.

### Slide 13

The latest improvement that I have incorporated into the solver is the ability to account for any geometric detail of the wind turbine.

By this, I mean that the control of tilt, yaw, azimuth angle, and pitch angle can all be modified and adjusted consistently during the time steps, which will offer the possibility to investigate several aspect of flows past wind turbines.

### Slide 14

To validate all the previous implementations and optimizations, the 5MW reference wind turbine designed by the National Renewable Energy Laboratory with a rotor radius 63m and a hub height of 90m is simulated in uniform laminar background flow operating at three tip speed ratio.

Each of the three cases consists of approximately 623 million cells, setting a new record in the ASCC supercomputers and uses around 4080 process for each computation.

### Slide 15

The focus is just to ensure that everything work well and that the overall topology of the flow is well restituted compared to previous studies, which is quite the case. Indeed, the generated helicoidal wake structure observed here through instantaneous contours of Qcriterion.

For the different tip speed ratios, a three vortex helicoidal structure is observed in the wake and the tip speed ratio induces a different spacing between the helicoidal vortex.

### Slide 16

To wrap things up, I have introduced a new solver that models wind turbines using the ALM method, and incorporated several improvement and optimization that can be used for more realistic configurations. The initailinitial results are promising.

The next steps involve parallelizing the ALM methods, creating a more comprehensive list of verification and validation, and then addressing the main problem, which is simulating an entire wind farm in the next year.