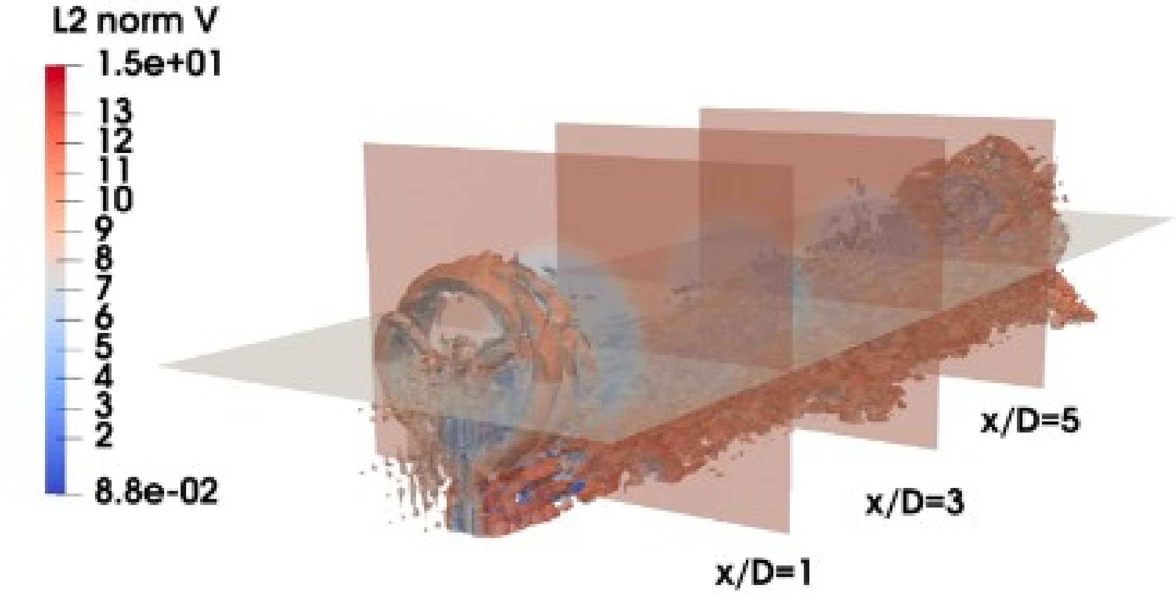
HORSES3D: Wind TurbiNe

A high-order discontinuous Galerkin solver for flow simulations and multi-physics applications



Universidad politécnica de madrid
Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio

1. Synopsis

HORSES3D is a multiphysics environment where the compressible Navier-Stokes equations, the incompressible Navier–Stokes equations, the Cahn–Hilliard equation and entropy–stable variants are solved. Arbitrary high–order, p–anisotropic discretisations are used, including static and dynamic p–adaptation methods (feature-based and truncation error-based). Explicit and implicit time-steppers for steady and time-marching solutions are available, including efficient multigrid and preconditioners. Numerical and analytical Jacobian computations with a coloring algorithm have been implemented. Multiphase flows are solved using a diffuse interface model: Navier–Stokes/Cahn–Hilliard. Turbulent models implemented include RANS: Spalart-Allmaras and LES: Smagorinsky, Wale, Vreman; including wall models. Immersed boundary methods can be used, to avoid creating body fitted meshes. Acoustic propagation can be computed using Ffowcs-Williams and Hawkings models.

HORSES3D supports curvilinear, hexahedral, conforming meshes in GMSH, HDF5 and SpecMesh/HOHQMesh format. A hybrid CPU-based parallelisation strategy (shared and distributed memory) with OpenMP and MPI is followed.

2. Installation

HORSES3D is an open-source code and it's available in GitHub: https://github.com/loganoz/horses3d

To download the code for a Unix-based OS, you can simply type in the terminal:

```
>> git clone https://github.com/loganoz/horses3d.git
```

As an alternative, you can download the zip file by clicking on the '<> Code' green button and then on 'Download ZIP'. To extract all directories, run:

```
>> unzip horses3d-master.zip
```

3. Compilation

HORSES3D is an object-oriented Fortran 2008 solver, that can be compiled using gcc and the Intel compiler, in Unix-based operating systems. We recommend using recent versions of such compilers (2019 or newer).

To compile the code, first go into the folder called *Solver*:

```
>> cd horses3d-master/Solver
```

Then, configure the project:

```
>> ./configure
```

Once the configuration is finished, you can build and compile the solver:

```
>> make clean

>> make all [options]
```

with the desired *options* (defaults are bold):

- PLATFORM=MACOSX/**LINUX**
- MODE=DEBUG/**RELEASE**
- COMPILER=ifort/**gfortran**
- COMM=PARALLEL/**SEQUENTIAL**
- ENABLE_THREADS=NO/**YES**
- WITH_PETSC=**NO/**YES
- WITH_METIS=**NO/**YES
- WITH_HDF5=**NO/**YES
- WITH_MKL=**NO/**YES

For example, to compile the code with the Intel compiler and MPI (parallel):

```
>> make all COMPILER=ifort COMM=PARALLEL
```

All flags that are nor specified will be compiled with the default values.

4. How to run a wind turbine simulation?

All details on how to run a simulation can be found in the **User Manual**, which is located in: **/horses3d-master/doc/UserManual.pdf**

We recommend following the cylinder tutorial before simulating the wind turbine.

First, go to the folder with the test case of the cylinder:

```
>> cd tutorials/Wind_Turbine
```

Inside that folder you will find:

- **Control file (wind_turbine.control)**: Inside that file you can specify all the parameters required to run the simulation.
- **RESULTS folder:** The output of the simulation will be stored inside that folder. If the folder doesn't exist, it will not be automatically created during the simulation and it will crash.
- **SETUP folder**: Inside this folder, there is a file called **ProblemFile.f90**, which allows you to select the initial condition and special boundary conditions, and to perform other operations during the simulation.
- **MESH folder**: Folder with the mesh of the domain.

In this tutorial the wind turbine is modelled using two combined approaches. The tower and nacelle are modelled through an immersed boundary method[1], while the rotating blades are modelled through and actuator line model.

[1] E. Ferrer, G. Rubio, G. Ntoukas, W. Laskowski, O.A. Mariño, S. Colombo, A. Mateo-Gabín, H. Marbona, F. Manrique de Lara, D. Huergo, J. Manzanero, A.M. Rueda-Ramírez, D.A. Kopriva, E. Valero, HORSES3D: A high-order discontinuous Galerkin solver for flow simulations and multi-physics applications, Computer Physics Communications, Volume 287, 2023, 108700

The geometry file in format ".stl" containing the tower and nacelle need to be located in the MESH/ folder. In addition, to use the actuator line model, a folder called "ActuatorDef" needs to be created. Within the latter there is a control file called "Act_ActuatorDef.dat", where the turbine is defined (including radius, rotational speed, blade section profiles, etc.) and also files containing the polars of the blade section profiles. Note that it is possible to define polar for different Reynolds numbers, and HORSES3D will interpolate to find the correct Reynolds number while running the simulation. *It is very important to not change the format or order of entries in these files.*

We simulate the experimental blind test (BT) of the Norwegian University of Science and Technology (NTNU), which is a three-bladed wind turbine with a diameter (D) of 0.894 m, and was tested in the wind tunnel of the Dept. Energy and Process Engineering, NTNU. The turbine blades were made up of NREL S826 airfoils[2]. The wind tunnel used for the experiment had dimensions of $[L \times W \times H]$ = [11.15 × 2.71 × 1.8] m. For our analysis, we used the setup and data from the blind test, which provided information on the wake of a single wind turbine at various tip speed ratios (λ). Specifically, we focus on the results at the optimal tip speed ratio $\lambda = \frac{\Omega D}{2U_{ref}} = 6$, with $U_{ref} = 10\frac{m}{s}$ and $\Omega = 134.228\frac{rad}{s}$, to evaluate the effectiveness of our adaptation methodology. The blade tip Reynolds number for this case is $\Re_c = \lambda U_{ref}$ and $\frac{c_{tip}}{v} = 103600$, where $c_{tip} = 0.025926\,m$ is the tip chord length and v is the kinematic viscosity of air.

A Cartesian mesh is generated with the same size as the wind tunnel, consisting of [128 × 24 × 24] elements. This results in a $\frac{D}{\Delta x}$ ratio of approximately 10, where D is the diameter of the turbine. It is important to note that, since we use a high-order method, the spatial resolution is also increased by raising the polynomial order.

For the wind turbine case, we need to solve the Navier-Stokes equations. Therefore, to start the simulation (from the Wind_Turbine folder) type:

```
>> ../../Solver/bin//horses3d.ns wind_turbine.control
```

To make things simple, a dynamic link of the executable file has been automatically created in the current folder. If it has been removed, you can create a new one with:

```
>> ln -s ../../Solver/bin//horses3d.ns
```

And then, start the simulation:

```
>> horses3d.ns wind_turbine.control
```

5. Post-processing

The results of the simulation will be stored inside the RESULTS folder in *.hsol files, which are binary files generated by HORSES3D. To visualize those results, we need to convert the results to a readable format *.tec, which can be visualize with the free CFD visualization tool **Paraview** (https://www.paraview.org/download/) or **Tecplot**.

[2] Per-Åge Krogstad and Pål Egil Eriksen. "blind test" calculations of the performance and wake development for a model wind turbine. Renewable energy, 50:325–333, 2013.

To convert the solution, run (from the Cylinder folder):

```
>> ../../Solver/bin/horses2plt solution_path.hsol mesh_path.hmesh [options]
```

The most important available options are:

- **--output-mode**: Either **multizone** or **FE**. The option multizone generates a Tecplot zone for each element. The option FE generates only one Tecplot zone for the fluid and one for each boundary. **FE is faster** in Paraview and Tecplot.
- **--output-variables**: Output variables separated by commas (see full list in the **User Manual**).

A complete list of the options is in the section 14.1 of the **User Manual**.

For the cylinder test case, let's convert the solution to show the density and the three components of the velocity:

```
>> ../../Solver/bin/horses2plt RESULTS/final_solution.hsol
MESH/final_mesh.hmesh --output-mode=FE --output-variables=rho,V
```

Note: "final_solution" and "final_mesh" must be changed to the names of the files with the final solution and the final mesh (the name depends on the number of iterations).

Again, you can also create a dynamic link to the executable file horses2plt:

```
>> ln -s ../../Solver/bin/horses2plt

>> horses2plt RESULTS/final_solution.hsol MESH/final_mesh.hmesh --output-
mode=FE --output-variables=rho,V
```

The conversion will generate a *.tec file inside the RESULTS folder, that can be opened from Paraview.