OpenNAOE Hull Resistance CFD

OpenNAOE Hull Resistance CFD (HullCFD) prepares a hull, builds and inspects its mesh, runs a fixed-hull calm-water resistance simulation with OpenFOAM, and reviews or exports the results. Version **1.0.0 - Windows and Linux**.

Install on Windows

Requires Intel/AMD 64-bit **Windows 10 build 19041 or later, or Windows 11**, and internet access to download missing components.

1. **Extract the complete ZIP** into a local folder. Keep its files together.
2. Double-click **INSTALL-WINDOWS.cmd**. Read the setup explanation and review the detected **Optimize WSL RAM and swap automatically** plan. This option is selected by default; it affects all WSL2 apps, preserves other settings and backs up an existing configuration. Tick **I understand...**, then click **Install and open**. For an existing installation, choose **Install update / repair**.
3. Approve any Windows administrator prompt. If a restart is requested, save your work and restart. If setup does not reopen, double-click **INSTALL-WINDOWS.cmd** after the restart, tick the acknowledgement and click the enabled install/repair button.
4. Open **HullCFD** from the desktop or the **OpenNAOE** Start menu folder.

Setup installs or reuses WSL2 and a dedicated **OpenNAOE-HullCFD** Linux environment with OpenFOAM v2412 (OpenCFD), Python and MPI. Personal Linux environments are kept separate.

The first installation can take tens of minutes. Installed simulations normally work offline.

If setup fails

Check the connection, then choose **Retry setup** or **Continue / repair**. Use **Show this attempt log** or **Startup diagnostics** to investigate. Keep the Linux environment and saved work in place. After cancelling, wait for the current installation operation to finish before starting another.

Install on Linux (Ubuntu)

The tool needs **OpenFOAM v2412** from OpenCFD (openfoam.com), **Python 3.12**, Open MPI and a C++ compiler (`g++`, `make`). Run setup as your normal user from the extracted folder:

```
bash install-linux.sh
```

- On **Ubuntu 24.04**, setup installs any missing system components (it asks for your administrator password), creates the application's Python environment and opens the app.
- On **another Linux system**, setup works only if all four requirements are already installed; it then reuses them and changes no system packages. The author runs the tool on such a system: Ubuntu 22.04 with OpenFOAM v2412 and Python 3.12, which Ubuntu 22.04 does not include and which he installed separately. This guide gives no installation steps for other systems.

To see what setup will find, run `bash packaging/linux-system.sh --check` before installing. It exits with 0 when compatible Python 3.12, OpenFOAM v2412, Open MPI and compiler tools are all present, and with 1 otherwise; it changes nothing.

To install OpenFOAM v2412 yourself on Ubuntu 24.04, use OpenCFD's repository, not a same-named package from another OpenFOAM distribution. Download and read the repository script before running it. This release uses OpenFOAM v2412 patch 260127, package version `2412.260127-1`, which setup also installs:

```
curl --fail --location --proto '=https' --tlsv1.2 \
  https://dl.openfoam.com/add-debian-repo.sh -o openfoam-repository.sh
less openfoam-repository.sh
sudo bash openfoam-repository.sh
sudo apt-get install openfoam2412-default=2412.260127-1
```

If your Ubuntu release offers another build of v2412, the run records name the build used and the hashes of its executables.

On every system, setup then installs the application itself: it copies it to `~/.local/share/OpenNAOE-HullCFD`, with its own workspace and a private Python 3.12 environment, and adds **HullCFD** to your applications menu. On a first installation it downloads the hash-locked Python packages of that environment, so internet access is needed; a later installation reuses a matching environment. If OpenFOAM v2412 is installed somewhere other than `/usr/lib/openfoam/openfoam2412` or `/opt/OpenFOAM/OpenFOAM-v2412`, set `SHIPCFD_OPENFOAM_BASHRC=/path/to/v2412/etc/bashrc` before running setup. Start the app later from the applications menu, or with `bash`

(C) 2026 Mohammed Abdul Hannan | Guide text: CC BY 4.0

`~/.local/share/OpenNAOE-HullCFD/opennaoe-hullcfd.sh`. The interface listens only on this computer (`127.0.0.1`).

Run a simulation

1. In **Hull**, load the example or import an STL/OBJ hull. Check the file units, scale ratio and bow direction.
2. In **Condition**, set draught, water properties and speeds. Review the waterplane and computed hydrostatics.
3. In **Mesh setup**, choose the mesh settings. Keep **Automatic** MPI selection to choose parallel workers from available RAM and CPU limits, or use **Manual**. Prepare the cases. **Coarse geometry check** is for inspecting geometry; use **Fine** or **Extra fine** for a resistance solve.
4. In **Build mesh**, build the selected case and inspect its mesh checks and saved 3D mesh. The mesh must pass its checks before solving.
5. In **Solve**, open **Solver settings**. Set **Simulation end time** and **Save fields every** in simulation seconds. The output interval must divide the end time exactly; for example, 2 s with output every 1 s. Choose **Update the current case** or **Create a new case using this mesh**, then click **Save solve settings**. Check the **Saved solve target** and click **Start fresh solve**.
6. After the solve finishes, open **Results**, select the saved case and click **Create analysis**. Select the saved analysis to review it, inspect its fields in **3D view**, and export the required outputs.

Saving settings retains earlier versions. **Start fresh solve** begins at 0 s; **Continue saved solution** extends existing output.

Automatic resource selection preserves the numerical mesh and solver settings. It cannot guarantee that an arbitrary mesh fits. If memory runs out, use **Save diagnostics** and the case's retry guidance. Increasing MPI workers does not add RAM. Existing cases retain their saved process count.

Save and find your work

The application stores cases, meshes and solver output in its Linux workspace:

- on Windows, `/home/opennaoe/.local/share/OpenNAOE-HullCFD/workspace` inside its dedicated WSL environment;
- on Linux, `~/.local/share/OpenNAOE-HullCFD/workspace`.

Use **Project file location** to open that workspace in File Explorer (on Linux, in your file manager).

- **Save setup** and **Save mesh settings** save the entered settings.
- **Save mesh** and **Save run status** save case records; actual mesh and solver files are already saved by the application.
- **Save project file** writes `OpenNAOE-HullCFD.hullcfd` in the workspace. **Open project file** restores its settings and references to saved cases.
- In **Results**, **Save results** saves CSV, history and report copies in the case folder. **Open results folder** opens that location. Export buttons download separate copies through the browser.

Saving opens the destination folder, or provides its path if the file manager cannot open. The save is retained.

Case folders are under `workspace/cases/`. Keep the Linux workspace in place: a `.hullcfd` project file references it and does not contain meshes or flow fields. Desktop folders can hold project-file copies and exports.

The named case/settings folder is a browsing index. Windows Explorer can show its `Inputs`, `Mesh` and `Solve` Linux links as small files. Read `Case-files.txt` there, or choose **Solve > Open solver files** to open the actual attempt. Serial field saves are numbered directories such as `0.05`; parallel saves are inside `processor0`, `processor1`, etc. The fixed-hull preset keeps the latest **five scheduled full-field saves per solver segment** and removes older ones. Force/residual histories are separate. **How it works** explains every case file, field, export and continuation step.

Remove or permanently delete a case

Remove case is available in the Build mesh and Solve case lists. It hides the case under **Removed from workspace** and keeps its files. **Restore case** returns it.

Delete permanently removes every saved version, mesh, solve, analysis and folder of that case. Shared source geometry, preparation history and external exports are kept. Deletion is blocked while the case is active or another case uses its mesh or saved solution. Review the confirmation before deleting.

(C) 2026 Mohammed Abdul Hannan | Guide text: CC BY 4.0

Reopen or continue a simulation

On Windows, open the **HullCFD** shortcut. If it cannot connect after a Windows restart, run **INSTALL-WINDOWS.cmd** and choose **Open existing app**. On Linux, open **HullCFD** from the applications menu.

Continue a saved solution

1. In **Solve**, select the case and choose **Continue saved solution**.
2. For an interrupted case, choose **Recover interrupted case** to verify that its old processes have stopped.
3. Click **Check complete saved writes**. Enter the additional simulation time and review the new total end time. For example, a saved write at **2 s** plus **3 s** gives a total end time of **5 s**.
4. Check the output interval, then click **Continue from ... to ...**.

Continuation retains the previous version and reuses its mesh and complete saved flow fields. Later interrupted progress is lost. Without a complete write, use **Start fresh solve** at 0 s with the accepted mesh.

Reopening leaves pending jobs paused; use the case controls to resume them. When finished, choose **Close application**. Closing only the window leaves background work running.

Update or uninstall

To update, finish active work and choose **Close application**. Extract the new ZIP into its own local folder, run **INSTALL-WINDOWS.cmd** and choose **Install update / repair**. Keep the existing Linux environment and workspace. On Linux, run `bash install-linux.sh` from the new folder; the workspace is kept.

To uninstall on Windows, close the application and run **UNINSTALL-WINDOWS.cmd**:

- **Remove launch icons only** removes the shortcuts and keeps the application and saved work.
- **Full uninstall** removes the listed dedicated Linux environments and application packages. It deletes **all projects and outputs inside those environments**, together with their Python, OpenFOAM and MPI components. Backup is selected by default; choose a backup folder outside the application and Linux disk folders. Confirm by typing **DELETE OpenNAOE**.

Shared Windows components, personal Linux environments, backups, logs, downloads and installation records are kept. Shared WSL RAM/swap settings are also kept; their original configuration backup is identified in setup diagnostics.

To uninstall on Linux, close the application, copy anything you want to keep from the workspace, then delete `~/.local/share/OpenNAOE-HullCFD` (this deletes the workspace and all its cases) and `~/.local/share/applications/opennaoe-hullcfd.desktop`. Setup leaves the system packages it installed and the OpenCFD package source it added. Remove packages that no other software needs with `apt`, for example OpenFOAM v2412:

```
sudo apt-get remove openfoam2412-default openfoam2412 \
  openfoam2412-common openfoam2412-tools
```

`apt` does not remove the package source. If no other software installs from the OpenCFD repository, delete its two files and refresh the package lists:

```
sudo rm /etc/apt/sources.list.d/opennaoe-openfoam.sources \
  /etc/apt/keyrings/opennaoe-openfoam.gpg
sudo apt-get update
```

Using the results

This tool supports learning, teaching and early design exploration of calm-water resistance by CFD, compared with model-test measurements for the KCS and DTC hulls. The numerical uncertainty of a run is not established; check the mesh, time step and convergence of each case, and qualify the tool further before real-world engineering use.

See **Verification and validation** and validation evidence (`docs/validation.md`).

Detailed controls and cluster runs: Using the application (`docs/local-gui.md`). Setup details: Installation, repair and removal (`docs/automatic-installer.md`).

Free to use under the **OpenNAOE Tool Licence** 1.0 (`LICENSE`): not for modification or redistribution; point others to opennaoe.org. The case templates in `templates/` and the cluster run scripts `scripts/run_mesh.sh` and `scripts/run_solve_time_accurate.sh` are under GPL-3.0-or-later; `LICENSING.md` explains the boundary, and `NOTICE` lists the third-party notices and where their licence texts are.

(C) 2026 Mohammed Abdul Hannan | Guide text: CC BY 4.0