
CFDEMcoupling Installation Guide

Release 6.3.0

DCS Computing GmbH

Nov 08, 2023

CONTENTS

1	Installation	3
1.1	Install Aspherix®	3
1.2	Install and compile OpenFOAM®	4
1.3	Install CFDEM®coupling	5
1.4	Test the installation	6
1.5	Visualization using Paraview and Macro	8
1.6	CFDEM Paraview reader	10
1.7	Advanced installation information (optional)	10



A Product of



CFDEM@coupling stands for Computational Fluid Dynamics (CFD)-Discrete Element Method (DEM) coupling.

CFDEM@coupling is an open-source code, distributed freely under the terms of the GNU Public License (GPL).

Core development of CFDEM@coupling is done by Christoph Goniva and Christoph Kloss, both at DCS Computing GmbH, 2012 (www.dcs-computing.com).

This documentation was written by DCS Computing GmbH, 2023.

INSTALLATION

This document will walk you through the installation process of CFDEM®coupling. First Aspherix® needs to be installed using its installer. OpenFOAM® and CFDEM®coupling are then installed. Finally, basic test cases are run to verify the successful installation.

As prerequisites OpenFOAM-10 and cmake ≥ 3.10 need to be installed. On Ubuntu 18.04+ cmake is installed during the OpenFOAM® installation with an appropriate version.

- *Install Aspherix®*
- *Install and compile OpenFOAM®*
- *Install CFDEM®coupling*
- *Test the installation*
- *Visualization using Paraview and Macro*
- *CFDEM Paraview reader*
- *Advanced installation information (optional)*

1.1 Install Aspherix®

Before installing CFDEM®coupling, it is necessary that Aspherix® is properly installed on your system. For more information regarding Aspherix®, please visit the [official website](#). You receive details of the Aspherix® installation together with the Aspherix® installer as part of the provided documentation. The default installation location of Aspherix® on your system is \$HOME/DCS-Computing. You can test your Aspherix® installation by typing `aspherix` in your console. If successful, this command should create the output

```
Aspherix (Version ...) Checkout of asx_solver OK.
```

1.2 Install and compile OpenFOAM®

Note: The current version of CFDEM®coupling compiles *only* with OpenFOAM® in version 10.

If do not want to compile OpenFOAM® yourself but prefer to install from a pre-compiled package, refer to [Install OpenFOAM® from package](#).

OpenFOAM is continuously updated and the source code repository contains numerous bugfixes that are not included in the pre-compiled package. To install and build the version that works best with CFDEM®coupling:

- As a prerequisite for the OpenFOAM® installation you need to download and install additional software packages (see the [OpenFOAM webpage](#) for more details). On Ubuntu 18.04 and newer execute the following command in a terminal:

```
sudo apt-get install build-essential cmake git ca-certificates flex libfl-dev bison
↳zlib1g-dev libboost-system-dev libboost-thread-dev libopenmpi-dev openmpi-bin
↳gnuplot libreadline-dev libncurses-dev libxt-dev
```

- Retrieve the source code. The following commands will create the required folders in an OpenFOAM directory in your HOME folder and download the desired version:

```
cd $HOME
mkdir OpenFOAM
cd OpenFOAM
git clone https://github.com/OpenFOAM/OpenFOAM-10.git
git clone https://github.com/OpenFOAM/ThirdParty-10.git
cd OpenFOAM-10
git checkout 362b7b6bb8eb5c4ac92e12856b3455cdacc1166a
cd ../ThirdParty-10
git checkout ae545a01baf66780285a2015da8a23c0a2dda5de
```

Alternatively, you can download the source code also as [zip archives](#).

- “Enable” the version to be compiled and used by loading the OpenFOAM environment variables in your `.bashrc` file. Open your `.bashrc` file from the terminal (here using the default text editor in Ubuntu):

```
gedit ~/.bashrc
```

and add the following lines *to the end* of the file.

```
#- OpenFOAM SETUP
export WM_NCOMPPROCS=8
source $HOME/OpenFOAM/OpenFOAM-10/etc/bashrc
```

Here, 8 is the number of the processors you want to compile with, adjust this number to as many cores as your system has or you are willing to use. Save the changes.

- To compile, open a new terminal (!) so that above changes to your `.bashrc` become active and execute the following steps:

```
cd $WM_PROJECT_DIR
./Allwmake
```

Depending on your hardware and the amount of cores used, compilation of OpenFOAM can take an hour or more.

- For testing your installation, test that the `simpleFoam` application is found. Type into your terminal

```
simpleFoam -help
```

and execute it. You should get a “Usage” message printed to screen.

1.3 Install CFDEM@coupling

You should have received a zip archive named “CFDEMcoupling-<PACKAGE>-<VERSION>” containing CFDEM@coupling, with PACKAGE being the package name and VERSION the version number. In the following examples we will assume usage of the Basic package. Perform the following to install CFDEMcoupling:

- Unzip the archive into the folder `$HOME/DCS-Computing`.

Note: Do not unzip the zip archive in Windows directly, but do this under Linux or from within the WSL. Since the Windows file system is not case sensitive, unzipping the archive directly in Windows may lead to missing files and, consequently, errors during compilation.

- Set required environment variables and load the CFDEMcoupling bashrc. To do so, add the following lines your `~/ .bashrc` and make sure to add them *after* loading the OpenFOAM bashrc:

```
#=====#
#- CFDEMcoupling SETUP
export CFDEM_RELEASE=|version|
export CFDEM_VERSION=Basic
export CFDEM_PROJECT_DIR=$HOME/DCS-Computing/CFDEMcoupling-$CFDEM_VERSION-$CFDEM_
↪RELEASE
export CFDEM_PROJECT_USER_DIR=$HOME/DCS-Computing/$LOGNAME-$CFDEM_VERSION-$CFDEM_
↪RELEASE
export CFDEM_bashrc=$CFDEM_PROJECT_DIR/src/lagrangian/cfdemParticle/etc/bashrc
export CFDEM_ASX_INSTALL_DIR=$HOME/DCS-Computing/Aspherix-$CFDEM_RELEASE
. $CFDEM_bashrc
#=====#
```

Setting the correct environment variables is crucial. In some cases you might need to deviate from the above template:

- The above example assumes that you are using the “CFDEMcoupling-Basic” version. If you use a different version, change the `CFDEM_VERSION` environment variable or modify other entries if necessary.
- If you are using a different Aspherix® version, e.g. a bugfix release, you may need to change the `CFDEM_ASX_INSTALL_DIR` variable accordingly.
- If you use c-shell, manipulate `~/ .cshrc` accordingly.

Note: As a shortcut or as a blueprint you can use the script `load_CFDEMcoupling-<PACKAGE>-<VERSION>.sh` located at the root of the zip archive.

If you directly load the environment using this script and do *not* update your `.bashrc`, you need stay in the same terminal and load this environment in every new terminal you want to use CFDEM@coupling in.

- Open a new terminal so that above changes to your `.bashrc` become active
- Test your setup using the system test command:

```
cfdemSysTest
```

Note: Please note the output of the number of threads per core and the recommendation for the CFD-DEM hyperthreading setting.

- Compile using

```
cfdemCompCFDEM
```

This command automatically compiles all the CFDEM@coupling libraries, CFDEM@coupling solvers and CFDEM@coupling utilities. The compilation will stop, if build errors occur.

The compilation is automatically logged and the logs can be found in:

```
$CFDEM_SRC_DIR/lagrangian/cfdemParticle/etc/log
```

Note: If compiling on a different Linux distribution than Ubuntu, *further settings* may be required.

In case questions concerning the installation arise, please feel free to contact [DCS Computing](#).

Note: The above installation instructions, assume you have already installed Aspherix® in the default location `$HOME/DCS-Computing`. Furthermore, we assume you have *installed OpenFOAM®*. If you are not using the CFDEM@coupling version `CFDEMcoupling-Basic`, please change the commands according to your CFDEM@coupling version.

1.4 Test the installation

A simple way to test your installation is by running a tutorial case. In order to run a single tutorial it is recommended to use the `Autorun.sh` scripts in the tutorial directories (or their older counterparts `Allrun.sh`). They execute the simulation and provide some basic output (e.g., comparison of simulation output with an analytic solution).

Now, we describe briefly how to run a simple test case – `cfdemSolverPiso/settling` – which models the settling of a single sphere using `cfdemSolverPiso` and plots its velocity.

Open a terminal and navigate to the tutorial folder:

```
cfdemTut
cd cfdemSolverPiso/settlingTest
```

Run the simulation by executing the `Autorun.sh` script in the terminal:

```
./Autorun.sh
```

The executed script will prompt to select a variant for the test cases to run:

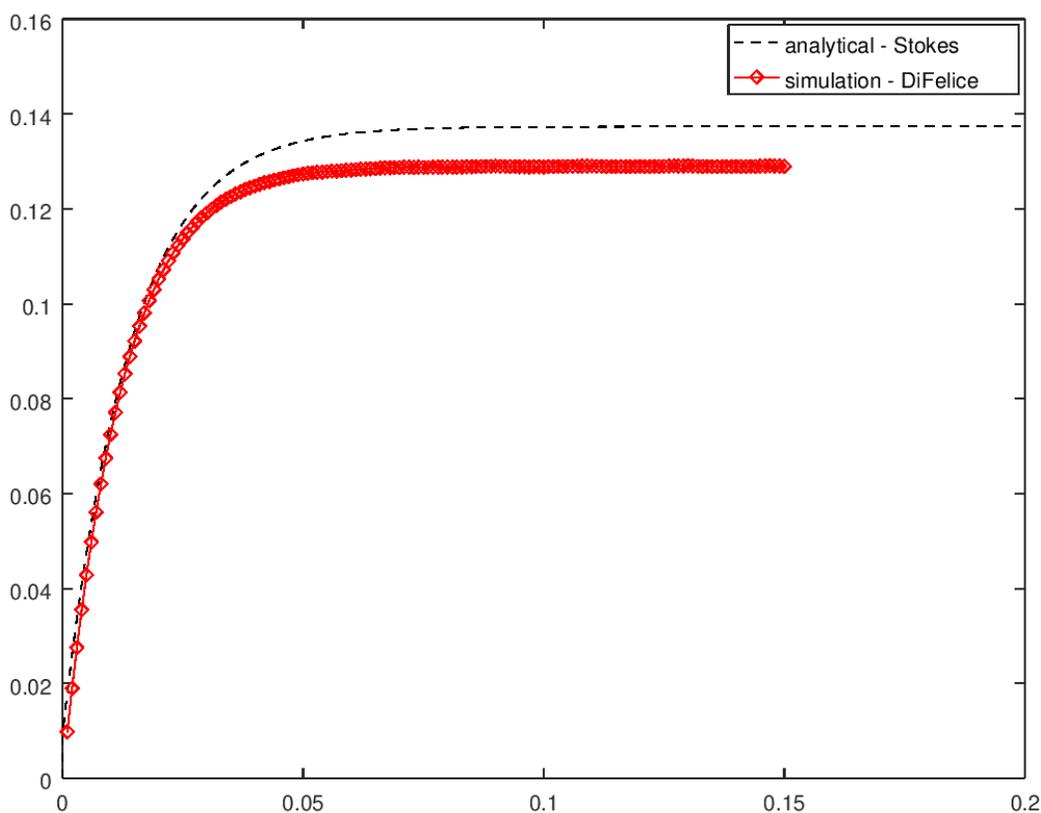
```
Available case variants (base variant is usually 'cfdemrun'):
```

- ```

1.) cfdemrun
2.) cfdemrun-serial
3.) cfdemrun-DEMBasedDrag
```

```
Which do you want to run? (Enter index):
```

Enter 1 to select the base variant `cfdemrun`. The other variants will change the case to use serial execution (2) or use a different drag law (for 3 the `DEMBasedDrag`). This will trigger Aspherix® to start the DEM simulation and CFDEM®coupling to start a coupled CFD-DEM simulation. Both simulations will communicate via socket and exchange the necessary data, e.g. particle position and velocity is communicated from DEM to CFD and the drag force is communicated from DEM to CFD. When the simulation is finished, a plot of the particle settling velocity pops up. The simulation result is compared to the analytical solution. You can visualize the simulation data using, e.g. Paraview.



In above plot the particle velocity (settling velocity) is plotted over time in red. The black line gives the analytical solution for the settling velocity using the Stokes drag law. Therefore, a slight offset between red and black line is to be expected.

You can run all the tutorial cases sequentially by executing the alias `cfdemTestTUT`. For running pure Aspherix® cases, you can use the aliases

```
cfdemAspherix inputScriptName
cfdemAspherixPar inputScriptName nOfProcs
```

In the tutorials section you find more details on setting up and running a tutorials.

---

**Note:** If you want to run your own cases, it is recommended to first copy the tutorial case to `$CFDEM_PROJECT_USER_DIR/run` which is automatically being generated. E.g., copy one of the tutorial cases there, and adapt it to your needs before running. This helps you keeping the tutorial in its original clean and working state.

---

## 1.5 Visualization using Paraview and Macro

To facilitate data loading a paraview macro is available to split loaded OpenFOAM / CFDEMcoupling data into separate blocks for internal field, patches, and –if present – particles.

---

**Note:** The macro is automatically installed during compilation. This means that you need to perform the steps described below only if you want to use the Macro on a system (or for a user) where no CFDEMcoupling is compiled.

---

### 1.5.1 Installation

To use this macro copy it from its location to the Paraview Macro folder:

```
cp -r $CFDEM_ETC_DIR/paraview/CFDEMcoupling* <pv_macro_dir>/
```

where the `pv_macro_dir` is:

- on Linux:  
  `$HOME/.config/ParaView/Macros'`
- on Windows  
  `%userprofile%\AppData\Roaming\ParaView\Macros`

On Linux (assuming the above location of the Paraview config folder) you can also use:

```
cfdemInstallParaviewMacro
```

### 1.5.2 Usage

Load the CFD results by opening any `*.foam` file in the CFD or `CFD/system` directories. After loading CFDEMcoupling simulation results you can run the `CFDEMcoupling` macro to split up the multiblock data, see Figure 2. Select **all** blocks when loading the data, i.e. also the boundaries and particles (if present), see Figure 1.

### 1.5.3 Particle data

By default, particle data is written by Aspherix and stored in the `DEM/post` folder. To load all particle data, simply load the file `aspherix_simulation.pvd` found in that directory.

To output particle data also on the CFD side, use an additional IOModel, e.g. `basicIO`.

---

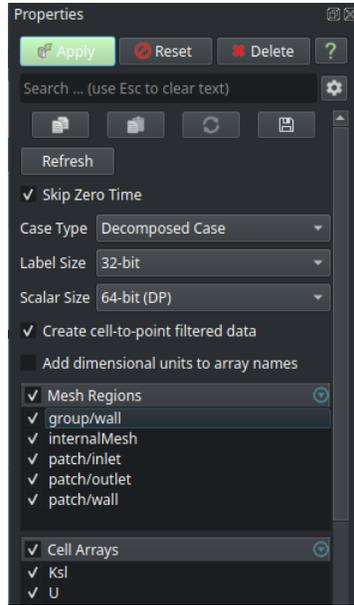


Fig. 1: Figure 1: Block selection.

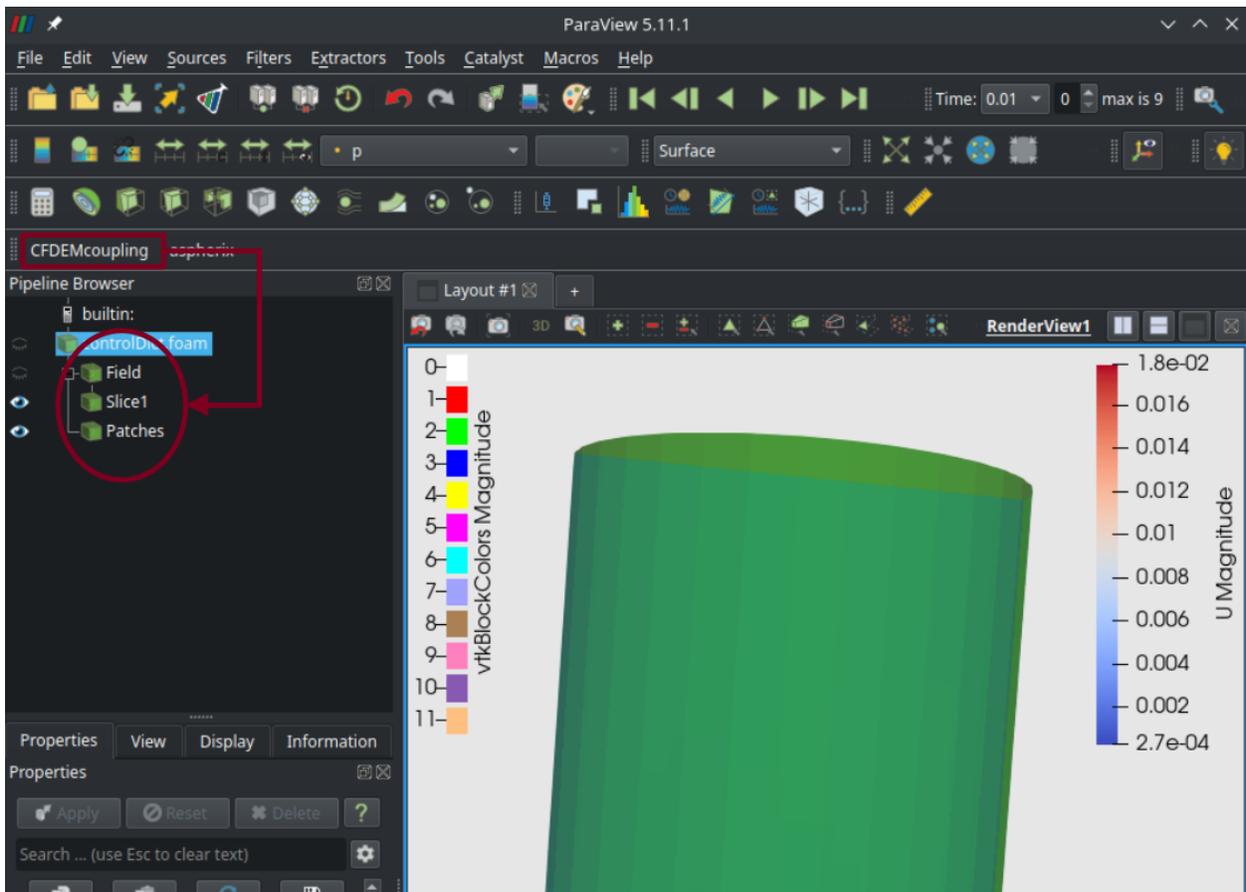


Fig. 2: Figure 2: CFDEMcoupling results split using the paraview macro.

## 1.6 CFDEM Paraview reader

In addition to the macro, a reader for coupled simulation results is installed automatically and should be available by default. If you install the macro manually as described above, the reader should also become available.

The CFDEMcoupling reader reads \*.cfdem files and exposes a node for, each, the CFD and the DEM results to which the corresponding macros may then attach. However, you still need to run the macros manually. The .cfdem file should be located in the root of the case directory.

---

**Note:** The reader is still at an early stage of development. If you encounter issues, please contact us.

---

---

**Note:** The usage of this reader is optional. You can always load the CFD and DEM results separately using the OpenFOAM and VTK-PVD readers.

---

## 1.7 Advanced installation information (optional)

In case the above instructions do not suit your needs, you can find advanced installation instructions in the following page:

### 1.7.1 Download OpenFOAM® zip archives

- Download the recommended OpenFOAM® version
  - **and** the corresponding ThirdParty software.
  - Unzip those archives to \$HOME/OpenFOAM and follow the other installation instructions.
- 

### 1.7.2 Install OpenFOAM® from package

---

**Note:** The current version of CFDEM®coupling compiles *only* with OpenFOAM® in version 10.

---

The easiest way to install *OpenFOAM®* on your system is by using the package manager to install a pre-compiled OpenFOAM package:

- As a prerequisite for the OpenFOAM® installation you need to download and install additional software packages (see the [OpenFOAM webpage](#) for more details). On Ubuntu 18.04 and newer execute the following command in a terminal:

```
sudo apt-get install build-essential cmake git ca-certificates flex libfl-dev bison ↵
↵zlib1g-dev libboost-system-dev libboost-thread-dev libopenmpi-dev openmpi-bin ↵
↵gnuplot libreadline-dev libncurses-dev libxt-dev
```

- Retrieve the OpenFOAM® package provided by OpenFOAM.org (find complete instructions for [Ubuntu](#)), which can easily be installed using the “apt” package manager. Basically the dl.openfoam.org repository must be added once to the list of software repositories. This is done with the terminal command:

```
sudo sh -c "wget -O - http://dl.openfoam.org/gpg.key | apt-key add -"
sudo add-apt-repository http://dl.openfoam.org/ubuntu
```

- Install the downloaded OpenFOAM® package using the terminal commands:

```
sudo apt-get update
sudo apt-get -y install openfoam10
```

- “Enable” the installed package by loading the OpenFOAM environment variables in your `.bashrc` file. Open your `.bashrc` file from the terminal (here using the default text editor in Ubuntu):

```
gedit ~/.bashrc
```

and add the following line *to the end* of the file.

```
source /opt/openfoam10/etc/bashrc
```

Save the changes.

- For testing your installation, open a new terminal (!) and test that the `simpleFoam` application is found. Type into your terminal

```
simpleFoam -help
```

and execute it. You should get a “Usage” message printed to screen.

### 1.7.3 OpenFOAM® and MPI

**Note:** This section discusses cases where a user-provided MPI is to be used with OpenFOAM. This is usually the case on clusters. On normal desktop systems, the default setting, i.e. `export WM_MPLIB=SYSTEMOPENMPI` should work and does not need to be changed.

Often problems arise configuring OpenFOAM® to work with a specific MPI installation on a cluster. However it has built-in capability to work with nearly any MPI implementation. Set the following variable in the OpenFOAM® `bashrc`:

```
export WM_MPLIB=SYSTEMMPI
```

Defining the following variables in your `~/.bashrc` *before* the sourcing of OpenFOAM® will take care of the rest. This is an example for older MVAPICH installations where there is no `libmpi.so`, only `libmpich.so` :

```
export MPI_ROOT=<path/to/mpi/installation>
export MPI_ARCH_PATH=$MPI_ROOT
export MPI_ARCH_FLAGS="-DMPICH_SKIP_MPICXX"
export MPI_ARCH_INC="-I$MPI_ARCH_PATH/include"
export MPI_ARCH_LIBS='-L$(MPI_ARCH_PATH)/lib -lmpich -lmpichcxx -lmpi -lopa -lrt'
```

## 1.7.4 Compiling CFDEM®coupling on different linux distributions

On some systems, like RHEL and CentOS, the MPICXX library is required in addition and you need to manually add it to the system linker paths *prior to compilation*. To do so, add

```
CFDEM_ADD_LIBS+="-lmpicxx"
CFDEM_ADD_LIBS+="-L/path/to/lmpicxx"
```

to

```
src/lagrangian/cfdemParticle/etc/addLibs_universal/additionalLibs_solver.
```

---

## 1.7.5 Compiling OpenFOAM® and CFDEM®coupling in debug mode

Set the WM\_COMPILE\_OPTION parameter in the OF bashrc to Debug and compile OF with the Allwmake script. CFDEM®coupling detects the OpenFOAM® variable and compiles in debug mode.

---

## 1.7.6 Install swak4FOAM (optional)

**Note:** Since migrating CFDEMcoupling to OpenFOAM-10, swak4FOAM is no longer used in the CFDEMcoupling tutorials. Consequently, the swak4FOAM installation is not tested with OpenFOAM versions beyond OpenFOAM-8.

---

swak4FOAM is a library of tools which open functionality otherwise only available by implementing new models. It is used in some test cases, mostly for post processing. To run these test cases without swak4FOAM, remove the related calls which are usually found in the functions subDict in controlDict.

To install swak4FOAM go to your OpenFOAM folder and run

```
clone repo into swak4Foam sub folder
git clone https://github.com/Unofficial-Extend-Project-Mirror/openfoam-extend-swak4Foam-
↳dev.git swak4Foam

go to cloned folder
cd swak4Foam
checkout version compatible to OpenFOAM-8
git checkout 0f570f524c0e3be44155cdf0f7623c5403f234d3

build
./AllwmakeAll
```

All dependencies and package requirements should already be installed from the package installation during the OpenFOAM installation process.

---

**Note:** If encountering compilation problems with swak4FOAM, try first to run the compilation on a single core by running `export WM_NCOMPPROCS=1` before invoking `./AllwmakeAll`.

---

---

**Note:** Refer to [swak4Foam](#) for further documentation on swak4FOAM.

---

### 1.7.7 Backward Compatibility

Basically, CFDEM®coupling supports one OpenFOAM® version. Therefore, all settings are prepared for that particular version. Nevertheless we try to maintain backwards compatibility as long as it works with reasonable effort.

The supported OpenFOAM® and Aspherix® versions are stated in:

```
src/lagrangian/cfdemParticle/cfdTools/versionInfo.H
```

For using other versions you can manipulate:

```
src/lagrangian/cfdemParticle/etc/OFversion/OFversion.H
```

### 1.7.8 Install git

Git allows you to update the source code very easily with the git pull command. On debian based systems you can run

```
sudo apt-get install git-core
```

Similar packages are available on other systems too:

```
sudo zypper install git-core
sudo yum install git
```

---

**Note:** In this tutorial the git protocol is used to transfer files. If port 9418 is closed for internet connections, please switch to “https://” instead of “git://” for git clone commands.

---

### 1.7.9 Install Helyx®OS (optional)

If you are interested in using a GUI based workflow for setting up your coupled CFD-DEM cases we recommend having a look into *Helyx®-OS* provided by *Engys*.

---

**Note:** This is to be considered only a first “proof of concept” of combining CFDEM®coupling and Helyx®OS, which is planned to be streamlined in the future.

---

The detailed installation instructions for Helyx®OS basically boil down to downloading the installer from this [link](#).

Then you need to change the permissions of the file by typing to the terminal

```
chmod 755 HELYX-OS-2.4.0-linux-x86_64.bin
```

And finally you need to execute the installer in a terminal by typing

```
./HELYX-OS-2.4.0-linux-x86_64.bin
```

**Note:** It is not required to install GUI + Kernel as you can use the OpenFOAM® version installed as described *above*. So please select only the components HELYX-OS-GUI. During installation the destination folder is being queried. As a default your user home directory is a good starting point. Simply type “echo \$HOME” in a terminal to see the path.

For a simple start of Helyx®OS we recommend to add to your ~/.bashrc (right above the definition of CFDEM\_VERSION) the path of your Helyx®OS installation (you can find the correct path by looking for the file HELYX-OS.sh):

```
export CFDEM_HELYXOS_DIR=$HOME/Engys/HELYX-OS/v2.4.0
```

In order to check your environment variables you can use the command

```
cfdemSysTest
```

, which will show your settings and if they are complete and correct. After defining CFDEM\_HELYXOS\_DIR you can start Helyx®OS by typing to your terminal:

```
cfdemHelyxOS
```

After the first start you will have to define the paths to your OpenFOAM® installation. Please add

```
/opt/openfoam6
```

to Edit/Preferences/Core Folder.

---

### 1.7.10 Install OpenFOAM on Windows

You can run OpenFOAM® and CFDEM®coupling natively on Windows using the Windows sub-system for Linux (WSL).

1. To prepare your Windows system, check that you have updated your Windows 10 to at least version 2004, e.g. displayed in *Settings -> System -> Info -> Windows specifications*. If you do not have this version installed, run the update from *Settings -> Update & Security -> Windows Update*.

---

**Note:** You can force the update of Windows 10 to version 2004 as described in [this manual](#).

---

2. Activate WSL by opening the PowerShell as Administrator via the start menu and execute the command:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

When prompted choose to restart Windows.

3. When rebooted, open a new PowerShell terminal with Administrator privileges and execute:

```
Enable-WindowsOptionalFeature -Online -FeatureName VirtualMachinePlatform
```

Again, when prompted choose to restart Windows.

- After the second reboot, start a new PowerShell terminal with Administrator privileges and to switch from WSL1 to the newer version WSL2, execute:

```
wsl --set-default-version 2
```

Now, WSL2 is set up and we proceed by installing a Ubuntu virtual machine which will be used by WSL.

- Start by opening the Microsoft Store and search for Ubuntu 20.04. Click *Install*. This will download the system image and may take some time depending on your internet connection.
- Start Ubuntu 20.04 from the start menu. This step will set up Ubuntu. In the process you must choose a username and a password for that user. These do not need to be identical with your Windows credentials.
- Within the Ubuntu system run the following commands to update the package repository and install required tools:

```
sudo apt update
sudo apt install build-essential
```

- OPTIONAL graphics support Install pre-requisites

```
sudo apt install nautilus gedit
sudo apt install xfce4
```

Open your bashrc (e.g. using nano)

```
nano .bashrc
```

and add the line

```
export DISPLAY=$(cat /etc/resolv.conf | grep nameserver | awk '{print $2}'):0
```

at the file end. Save the changes. If using nano, this is done using `Ctrl + o` followed by `Ctrl + x`.

- Create links to access the Windows file system more easily:

```
ln -s /mnt/c/Users/yourUserName/Downloads winDownloads
ln -s /mnt/c/Users/yourUserName/Documents winDocuments
```

- Close the terminal.
- Install [Paraview on Windows](#)
- OPTIONAL graphics support

To run graphical programs on the Linux machine you need to additionally install an Xserver on your Windows. For this purpose we recommend [VcXsrv](#). After installing VcXsrv, start it using the application Xlaunch.

---

**Note:** When doing so make sure to activate the checkbox “Disable access control”.

---

- OPTIONAL graphics support

Restart the ubuntu terminal.

In the terminal type

```
startxfce4 &
```

to start the graphical user interface.

Now, you are ready to proceed with the normal installation of *OpenFOAM* and *CFDEMcoupling* as described above within the just set up Ubuntu sub-system.

---

**Note:** After the installation of Aspherix-GUI you need to change the path of the Paraview installation to point to the one installed in Windows.

---

In case further questions concerning the installation arise, please feel free to contact [DCS Computing](#).

---

### Footnotes:

OPENFOAM® is a registered trade mark of OpenCFD Ltd, producer and distributor of the OpenFOAM software via <https://www.openfoam.com>.

HELYX® is a registered trade mark of Engys Limited, producer and distributor of the Helyx software via <https://engys.com>.