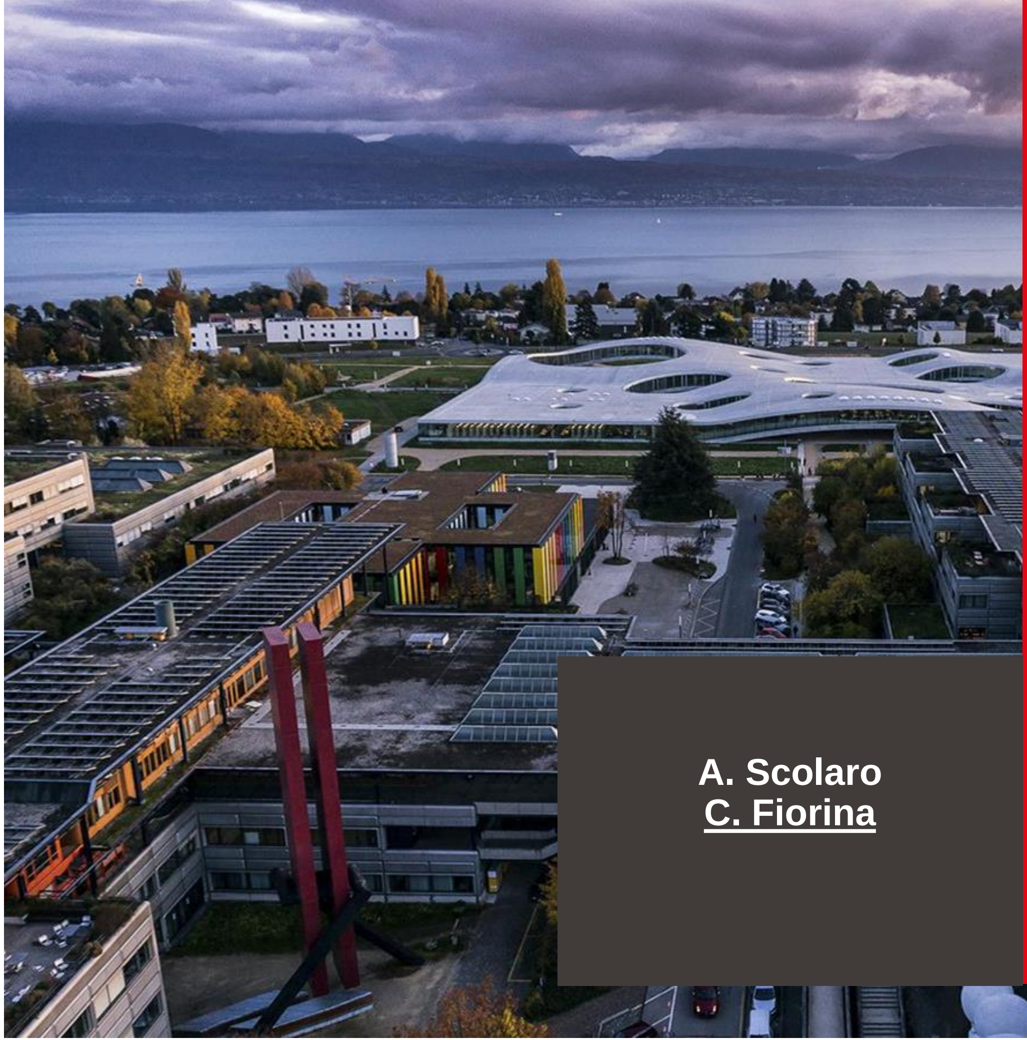
EPFL



Introduction to OpenFOAM

A. Scolaro
C. Fiorina

École polytechnique fédérale de Lausanne

I am curious about OpenFOAM … but which version?

OpenFOAM®

openfoam.***com***

OpenFOAM

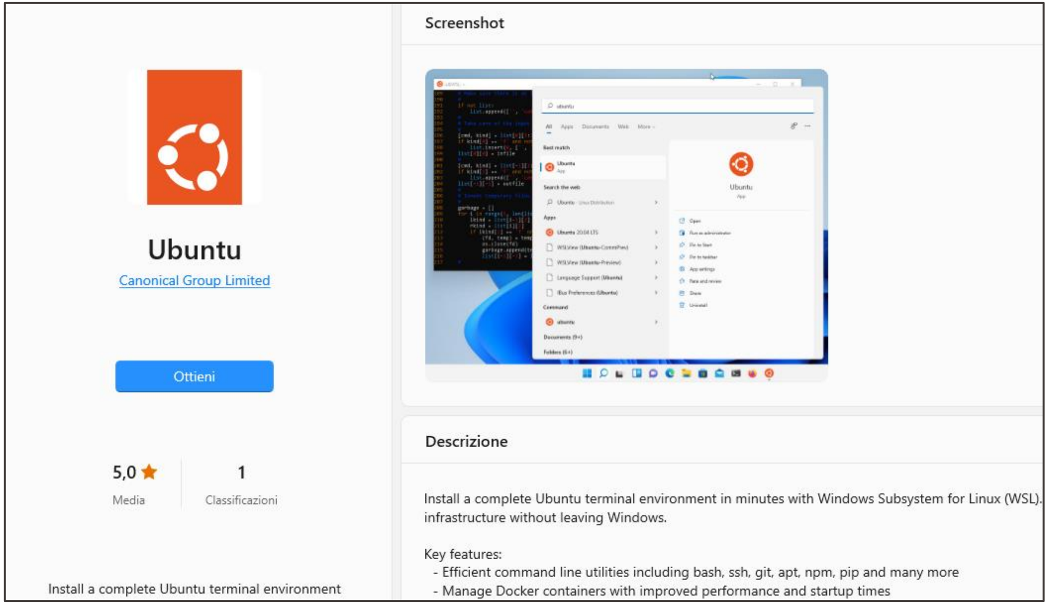
openfoam.***org***

IMPORTANT!

If you want to use an available solver, or take features from available solvers for your own solver, be very careful and select the right OF version!

Can I use it on my computer?

OpenFOAM runs natively on Linux systems…



MAC, or the Linux subsystem for Windows can be used, but **not recommended by the presenter**

How to get OpenFOAM?

Follow the simple steps on the download page

(example for OF-9 from the .org version)

Installation

OpenFOAM and *ParaView* can be simply installed for the first time using the **apt** package management tool. The user will need to provide superuser password authentication when executing the following commands with **sudo**

1. **Copy and paste** the following in a **terminal prompt** (*Applications → Accessories → Terminal*) to add **dl.openfoam.org** to the list of software repositories for **apt** to search, and to add the public key (**gpg.key**) for the repository to enable package signatures to be verified.
   Note: use secure **https://** **for the public key** to ensure secure transfer, but use**http://** **for the repository**, since **https://** may not be supported and is not required since the key provides secure authentication of the package files.

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

   ****Note: This only needs to be done once for a given system**

2. Update the **apt** package list to account for the new download repository location

   ```
   sudo apt-get update
   ```

3. Install OpenFOAM (9 in the name refers to version 9) which also installs **paraviewopenfoam56** as a dependency.

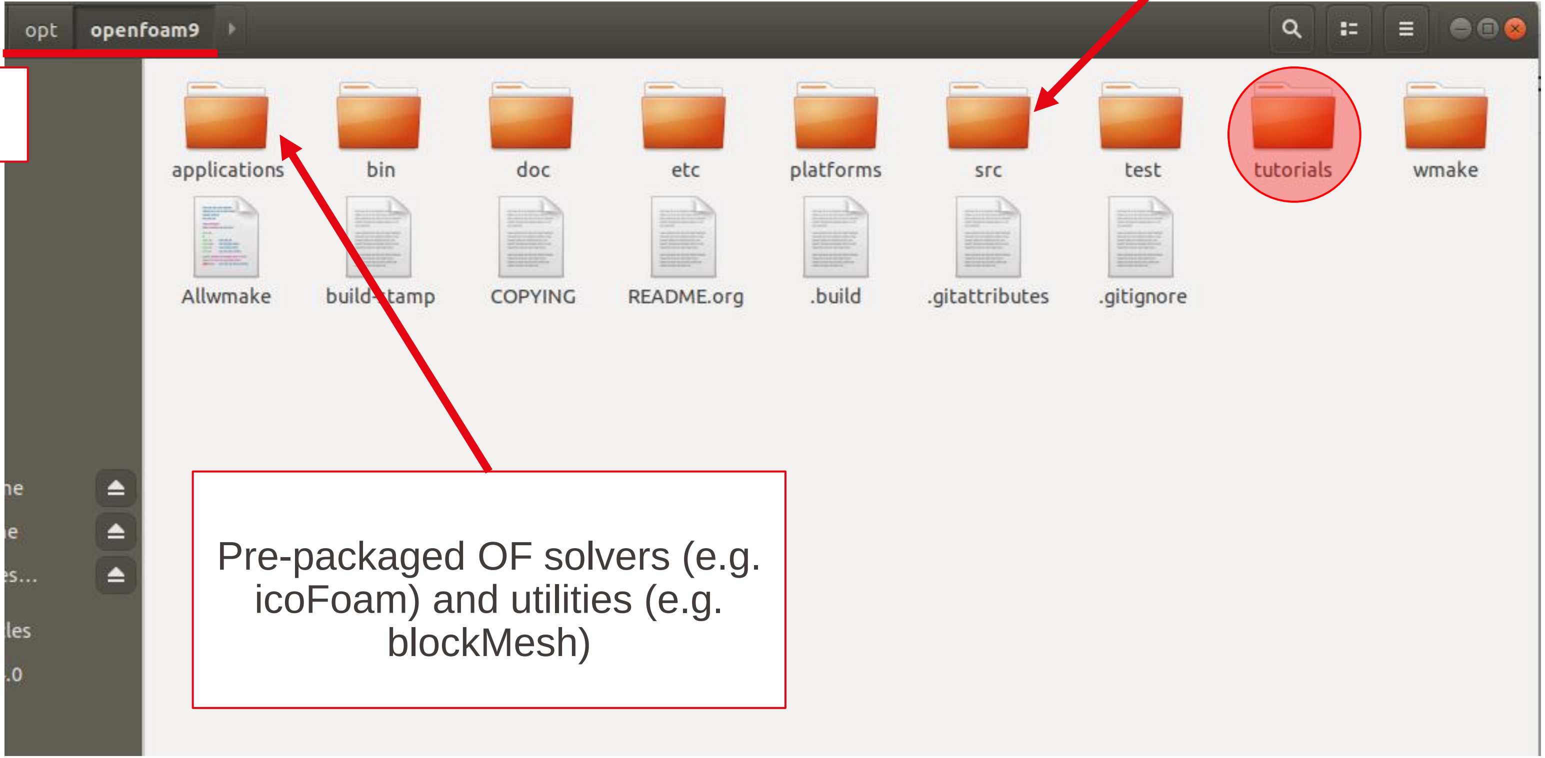
   ```
   sudo apt-get -y install openfoam9
   ```

OpenFOAM 9 and *ParaView 5.6.3* are now installed in the */opt* directory.

What comes with OpenFOAM?

Main OF library

Typical location



Pre-packaged OF solvers (e.g. icoFoam) and utilities (e.g. blockMesh)

Learn OpenFOAM - Official documentation

- https://cfd.direct/openfoam/user-guide/
- https://www.openfoam.com/documentation/user-guide

It includes some post-processing examples

CFD Direct
The Architects of OpenFOAM

Home Book OpenFOAM Cloud

OpenFOAM v9 User Guide: 2 Tutorials

[Table of Contents] [Index] [Version 9 | Version 8 | Version 7 | Version 6 | Version 5 | Version 4]

[prev] [next]

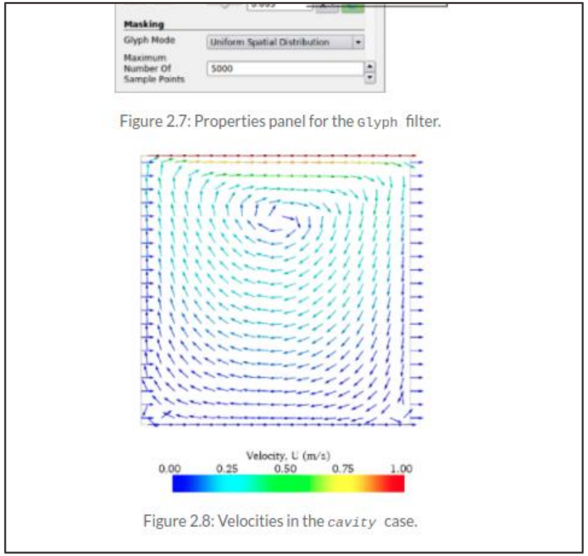
Chapter 2 Tutorials

In this chapter we shall describe in detail the process of setup, simulation and post-processing for some OpenFOAM test cases, with the principal aim of introducing a user to the basic procedures of running OpenFOAM. The `$FOAM_TUTORIALS` directory contains many more cases that demonstrate the use of all the solvers and many utilities supplied with OpenFOAM.

Before attempting to run the tutorials, the user must first make sure that OpenFOAM is installed correctly. Cases in the tutorials will be copied into the so-called *run* directory, an OpenFOAM project directory in the user's file system at `$HOME/OpenFOAM/<USER>-6/run` where `<USER>` is the account login name and "6" is the OpenFOAM version number. The *run* directory is represented by the `$FOAM_RUN` environment variable enabling the user to check its existence conveniently by typing

```
ls $FOAM_RUN
```

If a message is returned saying no such directory exists, the user should create the directory by typing



Learn OpenFOAM - Overview of Finite Volume Method from H. Jasak

https://www.youtube.com/watch?v=a4B_oXR5Kzs&ab_channel=KennethHoste

Diffusion Discretisation

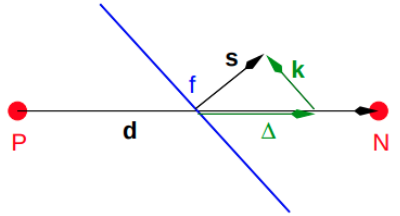
WIKKI

Diffusion Operator and Mesh Non-Orthogonality

- Diffusion term is discretised using the Gauss Theorem

$$\oint_S \gamma(\mathbf{n} \bullet \nabla \phi) dS = \sum_f \int_{S_f} \gamma(\mathbf{n} \bullet \nabla \phi)\, dS = \sum_f \gamma_f\, \mathbf{s}_f \bullet (\nabla \phi)_f$$

- Evaluation of the face-normal gradient. If $\mathbf{s}$ and $\mathbf{d}_f = \overline{PN}$ are aligned, use difference across the face. For non-orthogonal meshes, a correction term may be necessary



$$\mathbf{s}_f \bullet (\nabla \phi)_f = |\mathbf{s}_f| \frac{\phi_N - \phi_P}{|\mathbf{d}_f|} + \mathbf{k}_f \bullet (\nabla \phi)_f$$

Learn OpenFOAM - Take your time, follow the “3 weeks” series

https://wiki.openfoam.com/index.php?title=%223_weeks%22_series

3-weeks-series

Day 1	Day 2	Day 3	Day 4	Day 5
install - first steps	steps - visualization	introductory course	discretization	theory - fun simulations - tips
Day 6	**Day 7**	**Day 8**	**Day 9**	**Day 10**
geometry and meshing	turbulence 1	turbulence 2	multiphase	parallelization
Day 11	**Day 12**	**Day 13**	**Day 14**	**Day 15**
programming 1	programming 2	programming 3	programming 4	programming 5

EPFL

Learn OpenFOAM - Presentations from Wolf Dynamics

Running my first OpenFOAM® case setup blindf

Before we start – Always remember the directory structu

```
case_name
├── 0
├── constant
│   └── polyMesh
├── system
└── time_directories
```

- To keep everything in order, the case directory is often located in the path **`$WM_PROJECT_USER_DIR/run`**.
- This is not compulsory but highly advisable, you can put the case in any directory of your preference
- The name of the case directory if given by the user (do not use white spaces).
- You run the applications and utilities in the top level of this directory.
- The directory **`system`** contains run-time control and solver numerics.
- The directory **`constant`** contains physical properties, turbulence modeling properties, advanced phy and so on.
- The directory **`constant/polyMesh`** contains the polyhedral mesh information.
- The directory **`0`** contains boundary conditions (BC) and initial conditions (IC).

Solution initialization using codeStream

Body of the **codeStream** directive for initial conditions

```
internalField   #codeStream
{
    {
        codeInclude
        #{
            #include "fvCFD.H"
        #};

        codeOptions
        #{
            -I$(LIB_SRC)/finiteVolume/lnInclude \
            -I$(LIB_SRC)/meshTools/lnInclude
        #};

        codeLibs
        #{
            -lmeshTools \
            -lfiniteVolume
        #};

        code
        #{

        #};
    };
}
```

Initial conditions

Use codeStream to set the value of the initial conditions

Files needed for compilation

Compilation options

Libraries needed for compilation. Needed if you want to visualize the output of the initial conditions at time zero

Insert your code here. At this point, you need to know how to access internal mesh information

Learn OpenFOAM - Browse the C++ source guide official documentation

- https://www.openfoam.com/documentation/guides/v2112/doc/
- https://cpp.openfoam.org/v9/

- fixedGradientFvPatchField
- fixedInternalValueFvPatchField
- fixedJumpAMIFvPatchField
- fixedJumpFvPatchField
- FixedList
- fixedMeanFvPatchField
- fixedMeanOutletInletFvPatchField
- fixedMultiPhaseHeatFluxFvPatchScalarField
- fixedNormalInletOutletVelocityFvPatchVectorField
- fixedNormalSlipFvPatchField
- fixedNormalSlipPointPatchField
- fixedPressureCompressibleDensityFvPatchScalarField
- fixedProfileFvPatchField
- fixedRhoFvPatchScalarField
- fixedShearStressFvPatchVectorField
- fixedTrim
- fixedUnburntEnthalpyFvPatchScalarField
- fixedValueFvPatchField
- fixedValueFvsPatchField
- fixedValuePointPatchField
- flipLabelOp
- flipOp
- flowRateInletVelocityFvPatchVectorField
- flowRateOutletVelocityFvPatchVectorField
- fluentFvMesh
- fluidReactionThermo
- fluidSolutionControl

Detailed Description

template<class Type>
class Foam::fixedGradientFvPatchField< Type >

This boundary condition supplies a fixed gradient condition, such that the patch values are calculated using:

$$x_p = x_c + \frac{\nabla(x)}{\Delta}$$

where

x_p	=	patch values
x_c	=	internal field values
$\nabla(x)$	=	gradient (user-specified)
Δ	=	inverse distance from patch face centre to cell centre

Usage

Property	Description	Required	Default value
`gradient`	gradient	yes	

Example of the boundary condition specification:

```
<patchName>
{
    type            fixedGradient;
    gradient        uniform 0;
}
```

Learn OpenFOAM - Plenty of additional resources

- Tutorials/lectures (have a look on Google or YouTube)
- Master/PhD thesis etc.
- Forums
- (Often) direct communication with solver developers

And remember:

- **Don’t get frustrated: there is always a way out with OpenFOAM and, most likely, someone who had your same problem and will be happy to help**
- **Don’t get discouraged: the entry barrier may seem steep, but skills you’ll learn will allow you to tackle any kind of problems**
- **If possible, do not do it alone!**

Thank you

A. Scolaro
C. Fiorina