Disclaimer

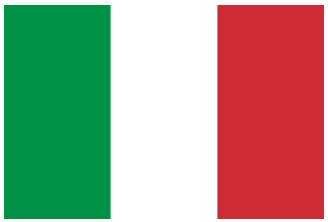
“This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks.”

Introductory OpenFOAM® Course

From 8th to 12th July, 2013

University of Genoa, DICCA

Dipartimento di **I**ngegneria **C**ivile, **C**himica e **A**mbientale



UNIVERSITÀ DEGLI STUDI
DI GENOVA

Your Lecturer

Joel GUERRERO

joel.guerrero@unige.it

UNIVERSITÀ DEGLI STUDI
DI GENOVA

guerrero@wolfdynamics.com

wolf dynamics

Acknowledgements

These slides and the tutorials presented are based upon personal experience, OpenFOAM® source code, OpenFOAM® user guide, OpenFOAM® programmer's guide, and presentations from previous OpenFOAM® training sessions and OpenFOAM® workshops.

We gratefully acknowledge the following OpenFOAM® users for their consent to use their material:

- Hrvoje Jasak. Wikki Ltd.
- Hakan Nilsson. Department of Applied Mechanics, Chalmers University of Technology.
- Eric Paterson. Applied Research Laboratory Professor of Mechanical Engineering, Pennsylvania State University.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

Today's lecture

1. How to implement a new boundary condition in OpenFOAM®
2. How to implement my own application in OpenFOAM®
3. How to add temperature to icoFoam

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

Today's lecture

1. **How to implement a new boundary condition in OpenFOAM®**
2. **How to implement my own application in OpenFOAM®**
3. **How to add temperature to icoFoam**

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to implement a new boundary condition in OpenFOAM®

Remember, all components in OpenFOAM® are implemented in library form for easy re-use. So, in order to implement a new boundary condition, we should follow these basic steps:

- The implementations of the boundary conditions are located in **$FOAM_SRC/finiteVolume/fields/fvPatchFields/**

- To add a new boundary condition, start by finding one that does almost what you want to do. Then, copy that boundary condition implementation to **$WM_PROJECT_USER_DIR** and modify it according to your needs.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to implement a new boundary condition in OpenFOAM®

Remember, all components in OpenFOAM® are implemented in library form for easy re-use. So, in order to implement a new boundary condition, we should follow these basic steps:

- Rename all the copied .C and .H files, for instance you can change their names to myFvPatchField. In the new files, search for all text occurrences with **originalFvPatchField** (the original name of the .C and .H files) and replace them with **myFvPatchField**.

- Modify the boundary condition to suit your needs. After you finish modifying it, compile it and use it as a dynamic library.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to implement a new boundary condition in OpenFOAM®

We will add a new parabolic inlet profile boundary condition.

- In **$path_to_openfoamcourse/programming/src/finiteVolume/fields/fvPatchFields/derived/parabolicVelocity** you will find the new boundary condition. Notice that we kept the same directory structure as in **$FOAM_SRC**.

From the terminal type:

- cd **$path_to_openfoamcourse/programming/src/finiteVolume/fields/fvPatchFields/derived/parabolicVelocity** (this is one line)

```
cd $path_to_openfoamcourse/programming/src/finiteVolume/fields/fvPatchFields/derived/parabolicVelocity
```

- wmake libso

If you did not get any error, the new boundary condition is ready to use. wmake will compile the new boundary condition and will place the new library in the directory **$WM_PROJECT_USER_DIR /platforms/linux64GccDPOpt/lib** (environment variable **$FOAM_USER_LIBBIN**).

Remember to modify the files located in the **Make** directory, so they reflect the location of the new library and its name.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to implement a new boundary condition in OpenFOAM®

Let us now setup a new case using the new boundary condition. From the terminal:

- cd **$path_to_openfoamcourse/mycases1/elbow2d_1/elbow_parabolic_inlet_0** (this is one line)

  cd $path_to_openfoamcourse/mycases1/elbow2d_1/elbow_parabolic_inlet_0

- Before running the solver we need to:
  - Add the new boundary condition parabolicVelocity in **0/**U.
  - Then add the line: **libs ("parabolicvelocityBC.so"),** to our controlDict dictionary in the **system** directory.

Then, from the terminal type:

- fluentMeshToFoam ../mesh/ascii.msh
  (the mesh is in the folder **../mesh**)
- checkMesh
- icoFoam
- paraFoam

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

A closer look at the newly created boundary condition

- The parabolicVelocityFvPatchVectorField boundary condition consists of two files:

  parabolicVelocityFvPatchVectorField.C
  parabolicVelocityFvPatchVectorField.H

- The .H file is the header file, and it is included in the header of the .C file.

- We can see that in the .H file we create a sub class to the fixedValueFvPatchVectorField:

  class parabolicVelocityFvPatchVectorField: public fixedValueFvPatchVectorField

  i.e. this is for Dirichlet boundary conditions for a vector field.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

A closer look at the newly created boundary condition

- The class has the private data

```
//- Peak velocity magnitude
scalar maxValue_;
//- Flow direction
vector n_;
//- Direction of the y-coordinate
vector y_;
```

- The TypeName("parabolicVelocity"), used when specifying the boundary condition, is defined.
- There are some public constructors and member functions that are defined in detail in the .C file.
- We used the third constructor when we tested the boundary condition, i.e. we read the member data from a dictionary.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

A closer look at the newly created boundary condition

- The actual implementation of the boundary condition can be found in the updateCoeffs() member function:

```
boundBox bb(patch().patch().localPoints(), true);
vector ctr = 0.5*(bb.max() + bb.min());
const vectorField& c = patch().Cf();
scalarField coord = 2*((c - ctr) & y_)/((bb.max() - bb.min()) & y_);
vectorField::operator=(n_*maxValue_*(1.0 - sqr(coord)));
```

- The member function write defines how to write out the boundary values in the time directory. The final line, writeEntry("value", os); writes out all the values, which is only needed for post-processing.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

A closer look at the newly created boundary condition

Find out more about all the variables by including the following lines at the end of the updateCoeffs member function:

```
Info << “c” << c << endl;
Info << “ctr” << ctr << endl;
Info << “y_” << y_ << endl;
Info << “bb.max()” << bb.max() << endl;
Info << “bb.min()” << bb.min() << endl;
Info << “(c - ctr)” << c - ctr << endl;
Info << “((c - ctr) & y_)” << ((c - ctr) & y_) << endl;
Info << “((bb.max() - bb.min()) & y_)” << ((bb.max() - bb.min()) & y_) << end;
Info << “coord” << coord << endl;
```

“This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks.”

Atmospheric boundary layer

Starting from the parabolic inlet profile boundary condition, let us now created a new boundary condition to simulated the atmospheric boundary layer.

In **$path_to_openfoamcourse/programming/src/finiteVolume/fields/fvPatchFields/derived/ablVelocity** you will find the new boundary condition. Take a look at the new boundary condition and identify the changes we have done. Also, try to figure out what kind of law we used to model the atmospheric boundary layer inlet profile. In the terminal type:

- cd **$path_to_openfoamcourse/programming/src/finiteVolume/fields/fvPatchFields/derived/ablVelocity**

cd $path_to_openfoamcourse/programming/src/finiteVolume/fields/fvPatchFields/derived/ablVelocity

- wmake libso

If you did not get any errors, the new boundary condition is ready to use. wmake will compile the new boundary condition and will place the new library in the directory **$WM_PROJECT_USER_DIR /platforms/linux64GccDPOpt/lib** (environment variable **$FOAM_USER_LIBBIN**).

Remember to modify the files located in the **Make** directory, so they reflect the location of the new library and its name.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to implement a new boundary condition in OpenFOAM®

Let us now setup a new case using the new boundary condition. From the terminal:

- cd **$path_to_openfoamcourse/mycases3/building/c0**

Before running the solver we need to:

- Add the new boundary condition ablVelocity in **0/**U.
- Then add the line: **libs ("ablvelocityBC"),** to your controlDict dictionary in the **system** directory.

Then, from the terminal type:

- fluent3DMeshToFoam ../mesh/ascii1.cas
  (the mesh is in the folder **../mesh**)
- checkMesh
- icoFoam
- paraFoam

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

Today's lecture

1. ~~How to implement a new boundary condition in OpenFOAM®~~
2. **How to implement my own application in OpenFOAM®**
3. How to add temperature to icoFoam

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to implement my own application in OpenFOAM®

Remember, all components in OpenFOAM® are implemented in library form for easy re-use. So, in order to implement a new application, we should follow these basic steps:

- The applications implementations are located in **$WM_PROJECT_DIR/applications**
- To add a new application, start by finding one that does almost what you want to do. Then, copy that application implementation to **$WM_PROJECT_USER_DIR** and modify it according to your needs.
- You can also write a solver from scratch. Remember to keep the prototype body of a solver.
- After you finish modifying or writing your own application, compile it and use it as you normally do.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to implement my own application in OpenFOAM®

Let us write a solver for the Laplace equation:

- In **$path_to_openfoamcourse/programming/applications/solvers/my_laplace/** you will find the new application.

From the terminal type:

- cd **$path_to_openfoamcourse/programming/applications/solvers/my_laplace/**
- wmake

If you did not get any errors, the new solver is ready to use. wmake will compile the new application and will place the new binary in the directory **$WM_PROJECT_USER_DIR /platforms/linux64GccDPOpt/bin** (environment variable **$FOAM_USER_APPBIN**).

Remember to modify the files located in the **Make** directory, so they reflect the location of the new library and its name.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to implement my own application in OpenFOAM®

So, what changes did we introduce in **my_laplace** directory:

- We created volumeScalarField T in createFields.H.
- We created the dimensionedScalar DT in createFields.H
- We added the Laplace equation in my_lapace.C

```
solve
(
    fvm::ddt(T) - fvm::laplacian(DT, T)
);
```

- Additionally we added in my_laplace.C the line:

```
#include "write.H”
```

to write out the volScalarField gradTx, volScalarField gradTx and volScalarField gradTx (the T gradient)

“This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks.”

How to implement my own application in OpenFOAM®

Let us now setup a new case using my_laplace. From the terminal:

- cd **$path_to_openfoamcourse/mycases0/mylaplace_1**

Before running the solver we need to:

- Add the new field variable file T to **0** directory.
- In fvSchemes, add the following line:

  laplacian(DT,T) Gauss linearUpwind grad(T);
- In fvSolution, add the solutions settings for T.

Then, from the terminal type:

- blockMesh
- my_laplace
- paraFoam

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

Today's lecture

1. ~~How to implement a new boundary condition in OpenFOAM®~~
2. ~~How to implement my own application in OpenFOAM®~~
3. **How to add temperature to icoFoam**

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

- Hereafter we will modify an existing solver (icoFoam).
- The modification will consist in adding temperature to the solver.
- We will need to copy the original solver into the directory **$WM_PROJECT_USER_DIR**. Remember to use the exact directory structure.
- After creating the new application's directory, various small changes are necessary to the solver files.
- Finally, the new field (temperature), needs to be added to the initial and boundary conditions. Also, we need to tell to OpenFOAM® how to discretize the new terms, this is done in the fvSchemes file. In fvSolution, we need to set the solver that we want to use to solve the field temperature.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

- In the folder **$path_to_openfoamcourse/programming/applications/solvers/icoFoam/,** you will find a copy of the original icoFoam solver. Let us copy the content of this folder to the folder **my_icoFoam.** From the terminal:
  - cp -r **$path_to_openfoamcourse/programming/applications/solvers/icoFoam/ $path_to_openfoamcourse/programming applications/solvers/my_icoFoam/** (this is one line)

```
cp -r $path_to_openfoamcourse/programming/applications/solvers/icoFoam/ $path_to_openfoamcourse/programming/applications/solvers/my_icoFoam/
```

- Now rename the file icoFoam.C to my_icoFoam.C.
  - mv icoFoam.C my_icoFoam.C

  Remember, the directory and the file must have the same name.

- Now go to the **Make** sub-directory, and open the file files with your favorite editor and add the following changes:

```
my_icoFoam.C
EXE = $(FOAM_USER_APPBIN)/my_icoFoam
```

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

- No changes are necessary for the file options.
- Now go back to the **my_icoFoam** directory. In the terminal,
  - cd ..
- Before compiling the new application, make sure that you do not have old binaries, old object files or old dependencies files in your application directory. In the terminal:
  - wclean

  This will clean the directory.
- Now we are ready to compile the new application. In the terminal:
  - wmake
- If everything went fine, your new solver binary should be in the **$FOAM_USER_APPBIN** directory. Check this with:
  - ls **$FOAM_USER_APPBIN**

You should see the new solver my_icoFoam in this directory.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

- So far we only compiled the application with a new name. Let us now add the necessary changes in order to solve the temperature field.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

- First, we need to create the new temperature field. Open the createFields.H file using your favorite editor.
- The first thing we can notice is that OpenFOAM® loads the kinematic viscosity from the transportProperties dictionary file. Let us add a new transport property related to the thermal diffusion which we will denote as DT. Make the following changes:

```
dimensionedScalar nu
(
    transportProperties.lookup("nu")        //Look for these lines
);
 //Starting from here add this line
dimensionedScalar DT
(
    transportProperties.lookup("DT")
);
//Done adding new lines
```

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

- If you keep reading the file, you will see that we also create the scalar field p (volScalarField) and the velocity field U (volVectorField). We need to add the new field for temperature T (volScalarField). After the creation of the field U and before the line #include “createPhi.H”, add the following lines:

```
Info<< "Reading field T\n" <<endl;
volScalarField T
(
    IOobject
    (
    "T",
    runTime.timeName(),
    mesh,
    IOobject::MUST_READ,
    IOobject::AUTO_WRITE
    ),
    mesh
);
```

“This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks.”

How to add temperature to icoFoam

- Save the modifications. At this point we have created the new field T and the new transport property DT.
- Let us now add the new equation to solve. Recalling that we want to add the transport equation for the scalar T (temperature) to the current solver

$$\frac{\partial T}{\partial t} + \nabla \cdot (\phi T) - \nabla \cdot (\nu \nabla T) = 0$$

Using OpenFOAM® equation mimicking pragma, we can write this equation as

```
fvScalarMatrix TEqn
(
        fvm::ddt(T)
    +   fvm::div(phi,T)
    -   fvm::laplacian(nu,T)
);
Teqn.solve( );
```

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

- Now, we need to add the new equation to the file my_icoFoam.C. Using your favorite editor add the following modifications.
- Because the temperature transport depends on the velocity field, we will add the equation after the momentum equation is solved (after the PISO loop), but before the time step is written. Edit your file so it looks like this:

```
        U -= rAU*fvc::grad(p);
        U.correctBoundaryConditions( );
    }

        //Starting from here add these lines
        fvScalarMatrix TEqn
        (
              fvm::ddt(T)
            + fvm::div(phi,T)
            - fvm::laplacian(nu,T)
        );
        Teqn.solve( );
        //End adding lines

        runTime.write();
```

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

- Save the modifications and compile the new solver. From the terminal
  - wclean
  - wmake

- If everything went fine, your new solver binary should be in the **$FOAM_USER_APPBIN** directory. Check this with:
  - ls **$FOAM_USER_APPBIN**

You should see the new solver my_icoFoam in this directory.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

- The next step is to test the new solver.

- Go to the directory **$path_to_openfoamcourse/mycases1/elbow2d_1/elbow_plus_temperature.** In the terminal:
  - cd **$path_to_openfoamcourse/mycases1/elbow2d_1/elbow_plus_temperature** (this is one line)

```
cd $path_to_openfoamcourse/mycases1/elbow2d_1/elbow_plus_temperature
```

- In this directory you will find the case setup files for a case using my_icoFoam solver.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

- Notice that we need to create the field file for T in the **0** directory (initial and boundary conditions).
- We also need to add the new transport property to the transportProperty dictionary.
- We need to tell to OpenFOAM® how to discretize the new equations. This is done in the fvSchemes dictionary.
- Finally, you will need to tell to OpenFOAM® how to solve the field T. This is done in the fvSolution dictionary.
- Take a look at these files (T, transportProperties, fvSchemes and fvSolution) and identify all the additions and changes made.
- **Always remember, OpenFOAM® is fully dimensional so your dimensions must be consistent.**

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

- We will now use the new solver. From now on follow me.
  - fluentMeshToFoam ../mesh/ascii.msh
    (the mesh is in the folder **../mesh**)
  - checkMesh
  - my_icoFoam
  - paraFoam

- Remember to go to the directory **$path_to_openfoamcourse/mycases1/elbow2d_1/elbow_plus_temperature.**

- In the **$path_to_openfoamcourse/mycases1/elbow2d_1/**, you will find various setup for the same geometry. For instance you could use my_icoFoam with a parabolic inlet boundary condition (**elbow_parabolic_inlet_plus_temperature**). Try to run all cases and identify the changes in the dictionaries.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

Remember, all components in OpenFOAM® are implemented in library form for easy re-use. So, in order to implement a new application, we should follow these basic steps:

- The applications implementations are located in **$WM_PROJECT_DIR/applications**
- To add a new application, start by finding one that does almost what you want to do. Then, copy that application implementation to **$WM_PROJECT_USER_DIR** and modify it according to your needs.
- You can also write a solver from scratch. Remember to keep the prototype body of a solver and implement your equations using OpenFOAM® equation mimicking pragma.
- After you finish modifying or writing your application, compile it and use it as you normally do.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

How to add temperature to icoFoam

At this point you should be able to modify OpenFOAM® solvers and create your own solvers and boundary conditions.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

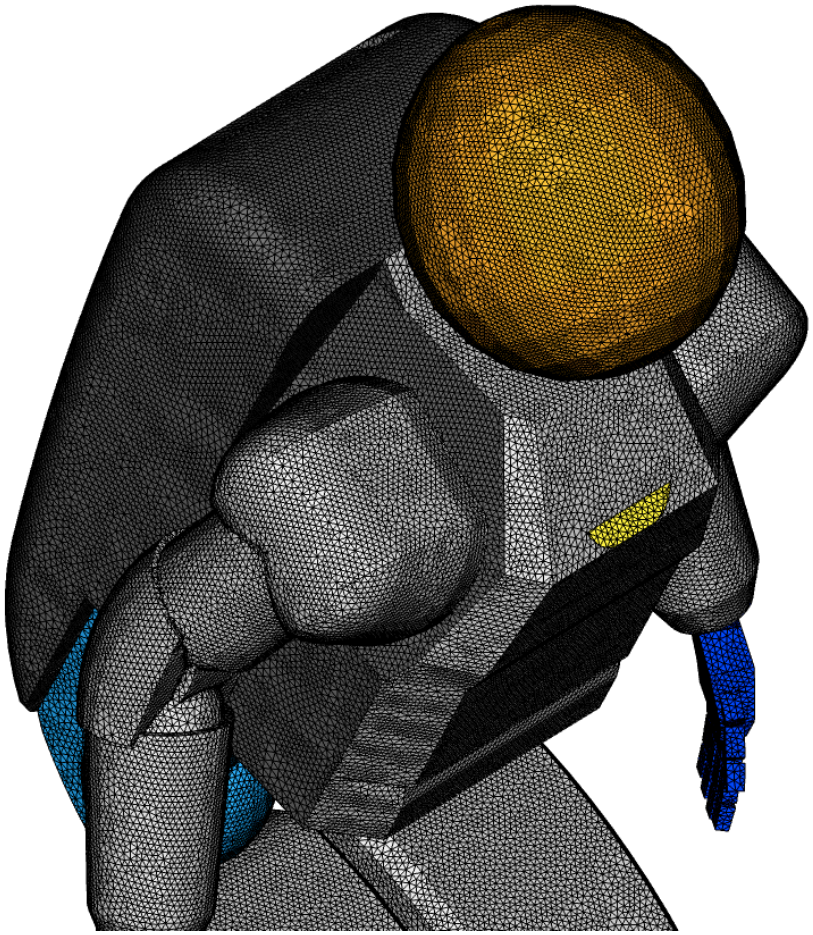
Mesh generation using open source tools

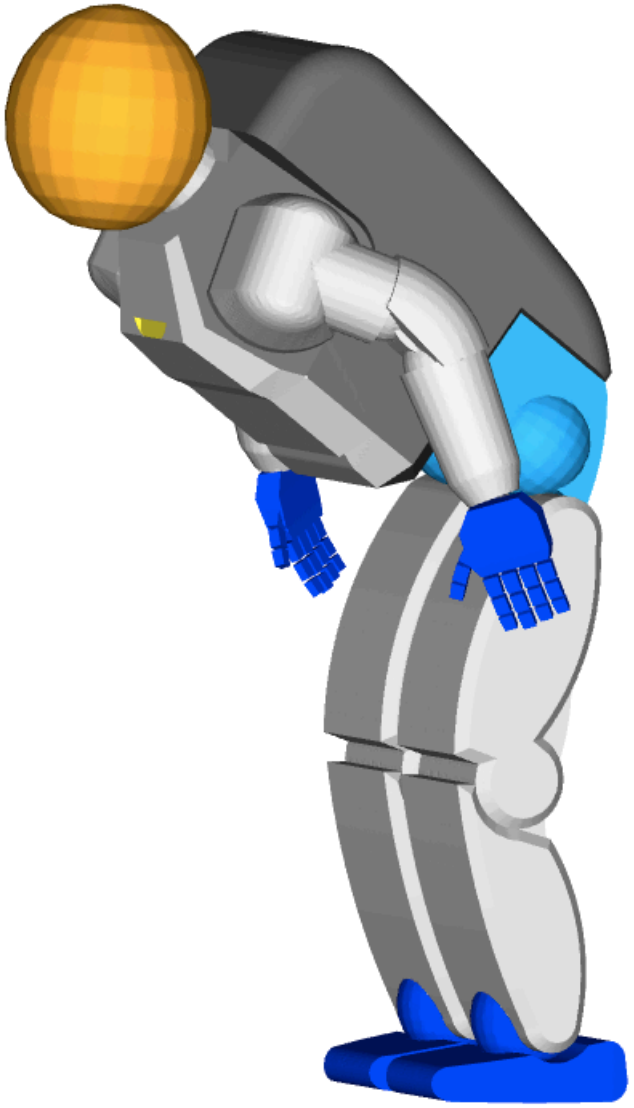
Additional tutorials

In the folders **$path_to_openfoamcourse/programming/applications** and **$path_to_openfoamcourse/programming/src** you will find many tutorials, try to go through each one to understand how to program in OpenFOAM®.

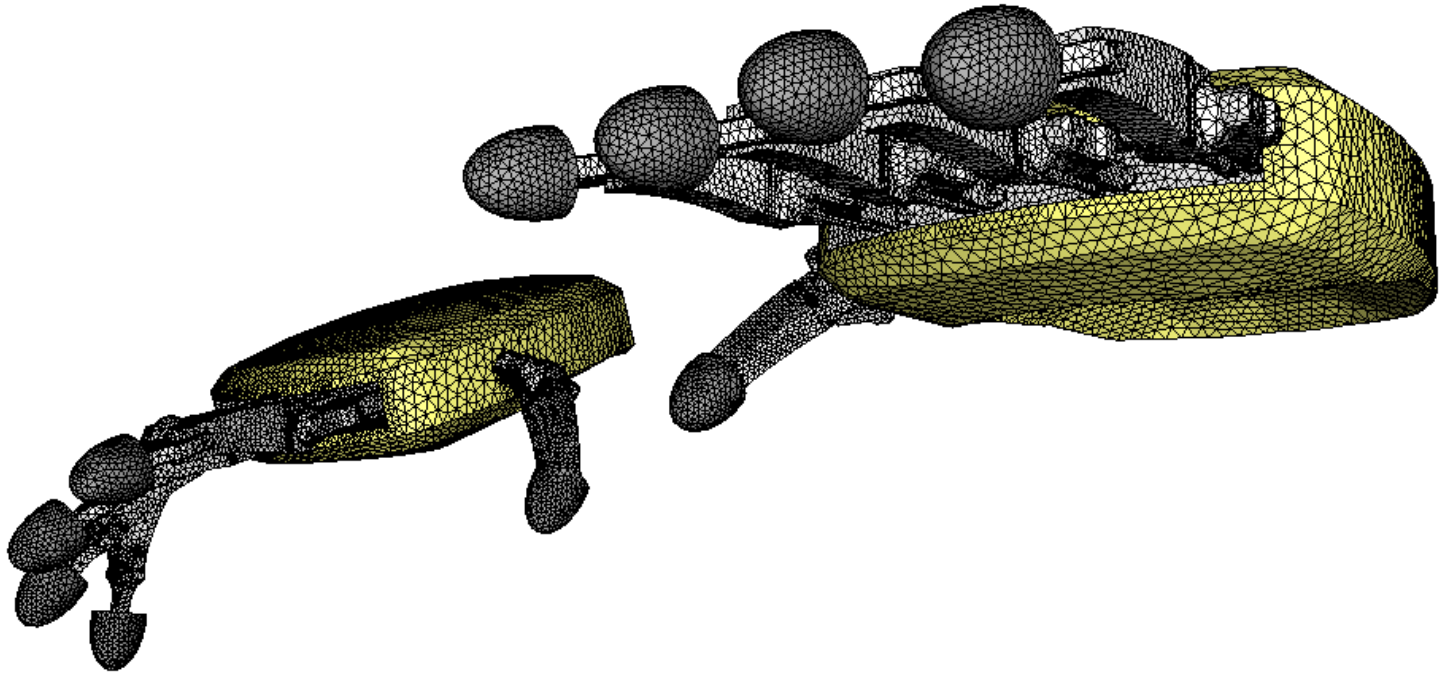
"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."

Thank you for your attention





Hands-on session



In the course's directory **($path_to_openfoamcourse)** you will find many tutorials (which are different from those that come with the OpenFOAM® installation), let us try to go through each one to understand and get functional using OpenFOAM®.

If you have a case of your own, let me know and I will try to do my best to help you to setup your case. But remember, the physics is yours.

"This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks."