

# OpenFOAM Tutorials: Programming Session

**Henrik Rusche**

`h.rusche@wikki-gmbh.de`

**Wikki, United Kingdom and Germany**

OpenFOAM-Workshop Training Sessions

7th OpenFOAM Workshop

24.6.2012, Darmstadt, Germany

## Working With OpenFOAM

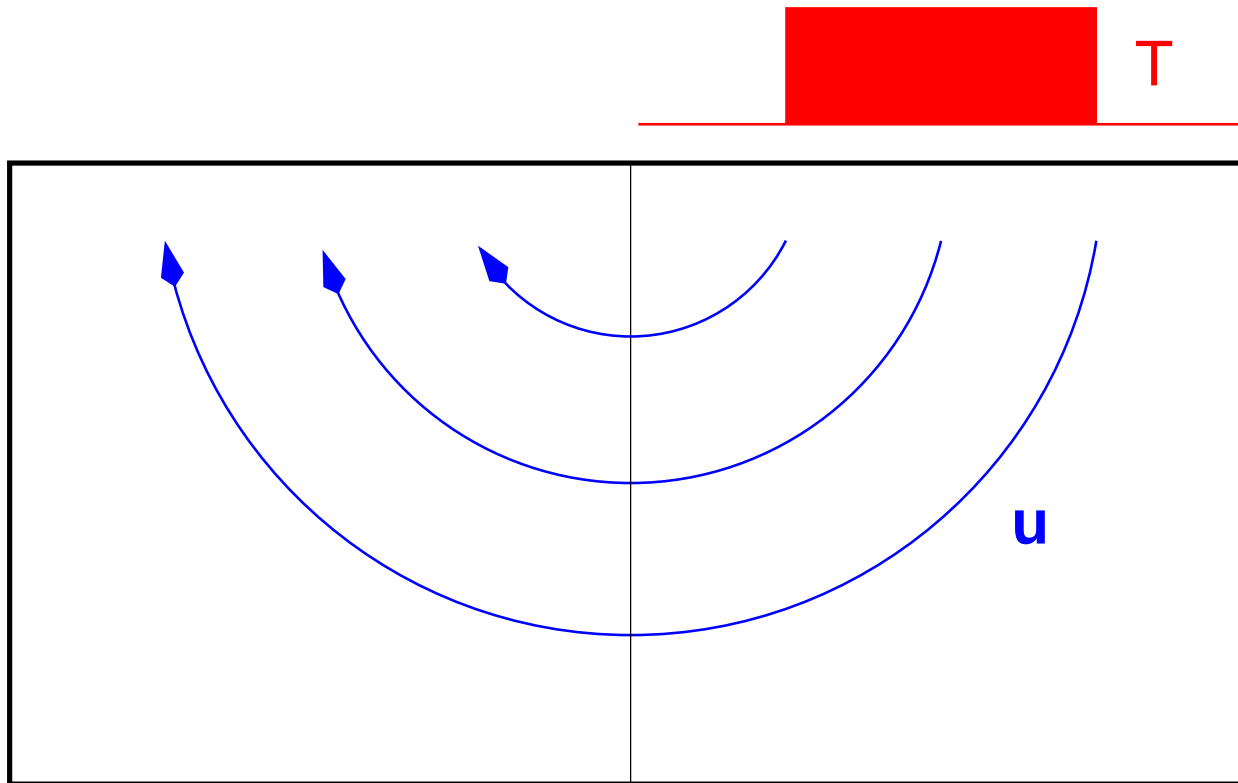
1. Walk through a simple solver: `scalarTransportFoam`
2. Scalar transport: swirl test
  - Non-uniform initial field
  - Field algebra and forced assignment
3. On-the-fly post-processing
4. Manipulating boundary values
5. Reading control data from a dictionary
6. Implementing a new boundary condition

## Solver Walk-Through `scalarTransportFoam`

- Types of files
  - **Header files**
    - \* Located before the entry line of the executable

```
int main(int argc, char *argv[])
```
    - \* Contain various class definitions
    - \* Grouped together for easier use
  - **Include files**
    - \* Often repeated code snippets, *e.g.* mesh creation, Courant number calculation and similar
    - \* Held centrally for easier maintenance
    - \* Enforce consistent naming between executables, *e.g.* `mesh`, `runTime`
  - **Local implementation files**
    - \* Main code, named consistently with executable
    - \* `createFields.H`

Swirl Test on `scalarTransportFoam`



- Setting up initial velocity field
- Forcing assignment on boundary conditions
- Types of boundary conditions

## Initial Condition Utility

```
volVectorField U
(
    IOobject
    (
        "U",
        runtime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::NO_WRITE
    ),
    mesh
);

// Do cells
const volVectorField& centres = mesh.C();

point origin(1, 1, 0.05);
vector axis(0, 0, -1);

U = axis ^ (centres.internalField() - origin);
U.write();
```

- swirlTest.git is a Git repository which contains snapshots of the input data and source code in its **branches**
- The slides have references to those branches
- Something like: In trouble? This is in branch **initU**
- With this information you can use

```
git checkout -f initU
```

to go to initU (throwing away your changes)

- or use

```
git checkout -f initU file1 file2 dir
```

to select individual files or directories

- or use

```
cd $HOME/swirlTest.git
```

```
rm -rf *
```

```
git checkout -f initU
```

to start at a given point. Note that you still have to compile and blockMesh!

- Unpack Git repo

```
cd $HOME  
tar xf /cdrom/OFW5/Advanced_Training/swirlTest.git.tgz  
cd $HOME/swirlTest.git
```

- Make the mesh

```
blockMesh
```

- Inspect the mesh
- Look at the U field
- You want to go back here? This is in branch **start**

- Compile setSwirl

```
cd $HOME/swirlTest.git/src/setSwirl  
wmake  
rehash
```

- Does it compile?
- Make the mesh and initialise the velocity field

```
cd $HOME/swirlTest.git  
setSwirl
```

- Does it run? Look at the U field! How did it change?
- In trouble? This is in branch **initU**
- Run scalarTransportFoam

```
scalarTransportFoam
```

- Does it run? Look at the result!



- Go there and compile

```
cd $HOME/swirlTest.git/src/postLib  
wmake libso
```

- Does it compile?
- Activate the minMaxField function object by uncommenting it

```
cd $HOME/swirlTest.git  
kate system/controlDict  
scalarTransportFoam
```

- Does it run? How did the output change?
- In trouble? This is in branch **minMax**

# Copy and rename scalarTransportFoam

- Copy scalarTransportFoam application from the OF-distro

```
cd $HOME/swirlTest.git/src
cp -r $FOAM_APP/solvers/basic/scalarTransportFoam .
```

- Rename it

```
mv scalarTransportFoam myScalarTransportFoam
cd myScalarTransportFoam
wclean
mv scalarTransportFoam.C myScalarTransportFoam.C
kate Make/files
wmake
rehash
```

- Make/files:

```
myScalarTransportFoam.C

EXE = $(FOAM_USER_APPBIN)/myScalarTransportFoam
```

- Does it compile? Does it run? In trouble? This is in branch **renamedApp**

- Add "Hello World"

```
kate myScalarTransportFoam.C  
wmake
```

- Add this line to myScalarTransportFoam.C:

```
for (runTime++; !runTime.end(); runTime++)  
{  
    Info<< "Time = " << runTime.timeName() << nl << endl;  
  
    Info<< "Hello World" << endl;  
  
    #        include "readSIMPLEControls.H"
```

- Does it compile?
- Does it compile? Does it run?
- In trouble? This is in branch **hello**

- Modify the source code and re-xcompile

```
kate myScalarTransportFoam.C  
wmake
```

- Modify myScalarTransportFoam.C such that:

```
for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)  
{  
    dimensionedScalar source  
    (  
        "source", dimensionSet(0, 0, -1, 1, 0), 1.0  
    ); // Added  
  
    solve  
    (  
        fvm::ddt(T)  
        + fvm::div(phi, T)  
        - fvm::laplacian(DT, T)  
        ==  
        source // Added  
    );  
}
```

- Does it compile? In trouble? This is in branch **uSource**

# Add a non-uniform source term (1)

- Add the field by copying the T field

```
cd $HOME/swirlTest.git
cp 0/T 0/source
kate 0/source
```

- 0/source:

```
dimensions      [0 0 -1 1 0 0 0];

internalField    uniform 1.0;

boundaryField
{
    fixedWalls { type zeroGradient; }

    inlet{ type zeroGradient; }

    outlet { type zeroGradient; }

    defaultFaces { type empty; }
}
```

- In trouble? This is in branch **copyField**

# Add a non-uniform source term (2)

- Initialise with a modified setSwirl utility

```
cd $HOME/swirlTest.git/src/setSwirl
kate setSwirl.C
wmake
setSwirl
```

- Add this section in setSwirl.C:

```
Info<< "      Reading source" << endl;
volScalarField source
(
    IOobject
    (
        "source",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::NO_WRITE
    ),
    mesh
);
```

- more on the next slide

# Add a non-uniform source term (3)

- Still in setSwirl.C:

```
...
U.write();

source.internalField() =
    centres.internalField().component(vector::X);

source.boundaryField()[0] ==
    centres.boundaryField()[0].component(vector::X);
source.boundaryField()[1] ==
    centres.boundaryField()[1].component(vector::X);
source.boundaryField()[2] ==
    centres.boundaryField()[2].component(vector::X);

source.write();
...
```

- Does it compile? Does it run? How did the source field change?
- In trouble? This is in branch **sourceField**

# Add a non-uniform source term (4)

- Modify the source code to use the source

```
cd $HOME/swirlTest.git/src/myScalarTransportFoam
kate myScalarTransportFoam.C
kate createFields.C
wmake
```

- createFields.H:

```
Info<< "Reading field source\n" << endl;
```

```
volScalarField source
(
    IOobject
    (
        "source",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```

- more on the next slide



# Add a non-uniform source term (5)

- Comment out one line in myScalarTransportFoam.C:

```
for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
{
    // Not needed!!!
    //dimensionedScalar source ("source", dimensionSet(0, 0, -1, 1, 0),

    solve
    (
```

- Does it compile?
- Does it run? Look at the result!
- In trouble: This is in branch **nuSource**

- Add the transport equation

```
cd $HOME/swirlTest.git/src/myScalarTransportFoam
kate myScalarTransportFoam.C
wmake
```

- Add 6 lines in myScalarTransportFoam.C:

```
for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
{
    solve
    (
        fvm::ddt(source)
        + fvm::div(phi, source)
        - fvm::laplacian(DT, source)
    );

    solve
    (
```

- Does it compile
- more on the next slide

- Change the boundary conditions and set it: Set inlet to fixedValue

```
kate 0/source  
setSwirl
```

- 0/source:

```
inlet  
{  
    type          zeroGradient;  
    type          fixedValue;  
    value         uniform 0;  
}
```

- Does it run? Did the BC in 0/source change?
- Now we run the application

```
cd $HOME/swirlTest.git  
myScalarTransportFoam
```

- Make the necessary changes in fvSchemes and fvSolution. Let the force guide you.
- In trouble? This is in branch **coupled**