Adding the scalar transport equation to icoFoam

- Let us modify a solver, we will work with **icoFoam**.
- We will add a passive scalar (convection-diffusion equation).
- At this point, you can work in any directory. But we recommend you to work in your OpenFOAM® user directory, type in the terminal,

```
1. $> cd $WM_PROJECT_USER_DIR/run
```

- Let us clone the original solver, type in the terminal,

```
1. $> cp -r $FOAM_APP/solvers/incompressible/icoFoam/my_icoFoam
2. $> cd my_icoFoam
```

- At this point, we are ready to modify the solver.

Adding the scalar transport equation to icoFoam

- Open the file *icoFoam.C* using your favorite editor and add the new equation in lines 118-123,

```
111                 U = HbyA - rAU*fvc::grad(p);
112                 U.correctBoundaryConditions();
113             }

118             solve
119             (
120                 fvm::ddt(S1)
121                 + fvm::div(phi, S1)
122                 - fvm::laplacian(DT, S1)
123             );

128             runTime.write();
129
```

Scalar transport equation.
The name of the scalar is S1.
We need to declare it in the createFields.H file.
We also need to read the coefficient DT.

- As the passive scalar equation depends on the vector field U, we need to add this equation after solving U.

Adding the scalar transport equation to icoFoam

- Open the file *createFields.H* using your favorite editor and add the following lines at the beginning of the file,

```
Info<< "Reading field S1 (passive scalar 1)\n" << endl;
volScalarField S1
(
      IOobject
      (
            "S1",
            runTime.timeName(),
            mesh,
            IOobject::MUST_READ,
            IOobject::AUTO_WRITE
      ),
      mesh
);

Info<< "Reading diffusionProperties\n" << endl;

IOdictionary diffusionProperties
(
      IOobject
      (
            "diffusionProperties",
            runTime.constant(),
            mesh,
            IOobject::MUST_READ_IF_MODIFIED,
            IOobject::NO_WRITE
      )
);

Info<< "Reading diffusivity DT\n" << endl;
dimensionedScalar DT
(
      diffusionProperties.lookup("DT")
);
```

Declaration of scalar field S1.
The solver will read the input file S1 (BC and IC).
You will need to create the file S1 in the time directory 0.

Declaration of input/output dictionary file.
The name of the dictionary is diffusionProperties and is located in the directory constant.

Read DT value from the dictionary diffusionProperties.

Adding the scalar transport equation to icoFoam

- Those are all the modifications we need to do.
- But before compiling the new solver, we need to modify the compilation instructions.
- Using you favorite editor, open the file *Make/files*,

Original file

```
icoFoam.C

EXE = $(FOAM_APPBIN)/icoFoam
```

Modified file

```
icoFoam.C

EXE = $(FOAM_USER_APPBIN)/my_icoFoam
```

Name of the input file

Name of the executable.
To avoid conflicts with the original installation, we give a different name to the executable

Location of the executable.
To avoid conflicts with the original installation, we install the executable in the user's personal directory

Adding the scalar transport equation to icoFoam

- At this point we are ready to compile, type in the terminal,

```
1. $> wmake
```

- If everything went fine, you should have a working solver named `my_icoFoam`.

- If you are feeling lazy or you can not fix the compilation errors, you will find the source code in the directory,

    - **`$PTOFC/101programming/applications/solvers/my_icoFoam`**

- You will find a case ready to run in the directory,

`$PTOFC/101programming/applications/solvers/my_icoFoam/test_case`

Adding the scalar transport equation to icoFoam

Running the case

- This case is ready to run, the input files are located in the directory **$PTOFC/101programming/applications/solvers/my_icoFoam/test_case**
- To run the case, type in the terminal,

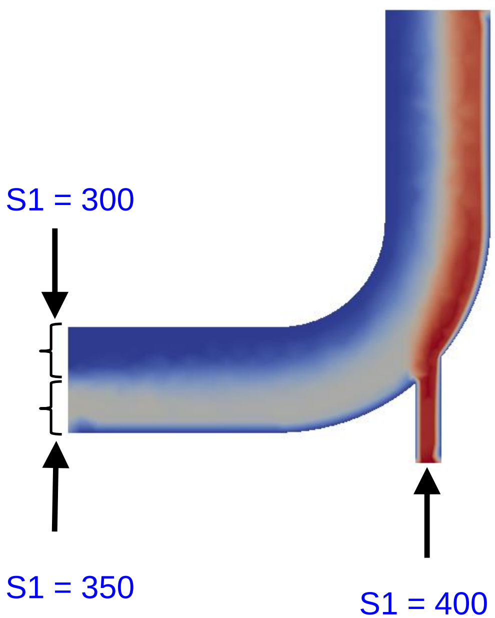
1. `$> foamCleanTutorials`
2. `$> fluentMeshToFoam ../../../../../meshes_and_geometries/fluent_elbow2d_1/ascii.msh`
3. `$> my_icoFoam | tee log`
4. `$> paraFoam`

- Remember, you will need to create the file *0/S1* (boundary conditions and initial conditions for the new scalar).
- You will also need to create the input dictionary *constant/diffusionProperties*, from this dictionary we will read the diffusion coefficient value.
- Finally, remember to update the files *system/fvSchemes* and *system/fvSolution* to take into account the new equation.

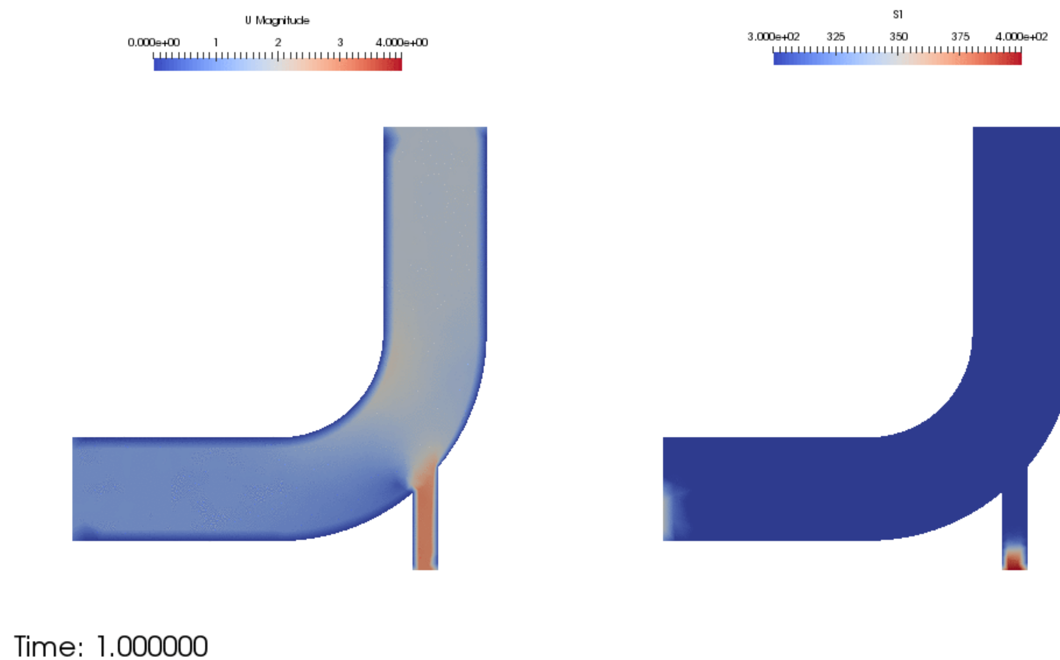
Adding the scalar transport equation to icoFoam

Running the case

- If everything went fine, you should get something like this



S1 inlet values



Visualization of velocity magnitude and passive scalar S1

www.wolfdynamics.com/wiki/BCIC/2delbow_S1