



Lab Migration Report

On

**Modelling And Simulation Of The Transport Of Passive
Scalar In Laminar Flow As Tracer With Diffusion**

Proposed By:

Prof. Dhiraj Garg

Solution By:

John Pinto

Manjil Sitoula

Table of Contents

1. Governing Equations and Models	3
1.1 Problem Statements.....	3
1.2 Governing Equations.....	3
1.3 Geometry and Mesh.	3
1.4 Solver Setup	4
1.4.1 Fluid Properties	4
1.5 Case Setup	5
1.6 Initial and Boundary Conditions	Error! Bookmark not defined.
1.6.1 Flow Field	8
1.6.2 Scalar.....	9
2. Results and Discussions.....	10

List of Figures

Fig 1: Schematic Diagram of Computational Domain	4
Fig 2: Mesh Generation using BlockMesh	4
Fig 3 Tree diagram of the main folder	5
Fig 4 Detailed Tree diagram of the directories	6
Fig 5 Scalar Concentration along the tube at various time steps	12
Fig 6 Scalar concentration at the outlet.....	13
Fig 7 Scalar concentration vs time at the Outlet.....	13

List of Tables

Table 1-1 Boundary Conditions for U	8
Table 1-2 Boundary Conditions for p	8
Table 1-3 Boundary Conditions for scalar.....	10

1. Governing Equations and Models

1.1 Problem Statements

To model and simulate the transport of passive scalar in laminar flow as tracer (non-reactive chemical species) with diffusion .

Objective – The tracer would be injected into upper half of the circular cylinder inlet. The flow will be maintained in laminar range to avoid any radial flow. Between the two mechanisms (convection and diffusion) for mass transfer, radial convection is eliminated by ensuring laminar flow in straight cylindrical tube. Thus, the only way a tracer can move in a radial direction is through diffusion alone. The higher is the diffusion, higher will be the movement. This is what we want to observe through this exercise.

1.2 Governing Equations

The continuity and momentum conservation equations are solved for the calculation of flow parameters like velocity and pressure. The conservation equations are expressed below:

Continuity Equation:

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

Momentum Equation:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} (\nabla p) + \nu \nabla^2 \mathbf{u} \quad (2)$$

where ν is the kinematic viscosity, ρ is the density, \mathbf{u} is the velocity vector and p is the pressure.

A concentration transport equation is incorporated for the calculation of passive scalar concentration using modified solver.

Equation to model the transport of Passive Scalars (S1):

$$\frac{\partial S}{\partial t} + \mathbf{u} \cdot \nabla S = \nabla \cdot (\Gamma \nabla S) + S_C \quad (3)$$

Where S is the passive scalar, Γ is the diffusion coefficient and S_C is the source term of the passive scalar. The source term would be required to model chemical reaction related to S which would be 0 here. For more information regarding the source term, you can refer to this document [Theory](#).

1.3 Geometry and Mesh.

The domain is a straight cylindrical tube with length of 0.5 m and diameter of the cross section as 0.01 m as shown in Fig. 1. The geometry is 3D. The geometry is long enough for the flow to fully develop. The tracer will be injected at upper half of the inlet.

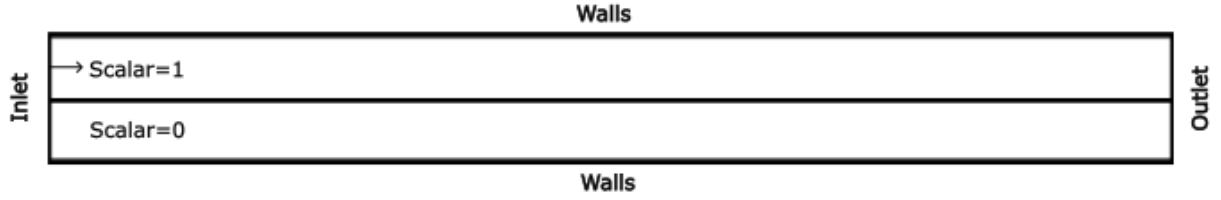


Fig 1: Schematic Diagram of Computational Domain

The meshing is done using blockMesh utility, openFOAM's built-in tool. The geometry is divided into 5 blocks and 18 vertices and the number of cells is 672000 for this case. The user is free to choose different types of meshing and numbers to get the similar results. For the detailed meshing process, one can go through the openFOAM spoken tutorial number 6. [spoken tutorial](#)

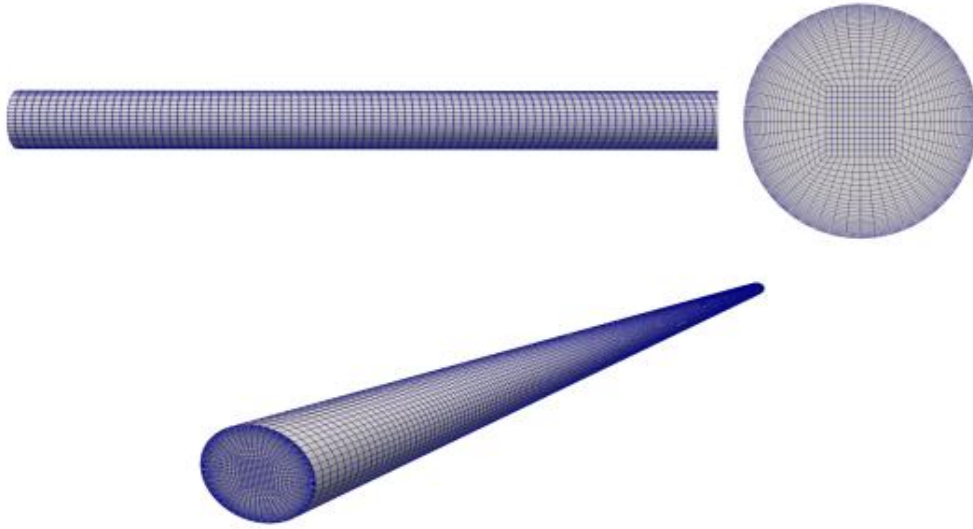


Fig 2: Mesh Generation using BlockMesh

1.4 Solver Setup

1.4.1 Fluid Properties

Water at room temperature is used as the fluid and the thermo physical properties of water are taken for calculations. The kinematic viscosity of water at room temperature (300 K) is $8.58 \times 10^{-7} \text{ m}^2/\text{s}$. The average velocity of the flow is 0.1 m/s , which is half the maximum velocity. The Reynolds number is expressed as:

$$Re = \frac{U_{avg} * D}{\nu}$$

Where U_{avg} is the average velocity, D is the diameter of the tube and ν is the kinematic viscosity. For the above input flow parameters, the Reynolds number of the flow is 1165 which is in the laminar regime (<2100). So, a laminar model is used for the simulation.

Flow Parameters	Value
Max. Velocity (U_{max})	0.2 m/s

Average Velocity (U_{avg})	0.1 m/s
Density (ρ)	1000 kg/m ³
kinematic viscosity (ν)	$8.58 \times 10^{-7} \text{ m}^2/\text{s}$
Reynolds No.	1165
Scalar Diffusivity constant (DS1)	$1 \times 10^{-6} \text{ m}^2/\text{s}$
Scalar Kinetic rate coefficient(kS1)	0 s^{-1}
S_c	-kS1*S1

1.4.2 Case Setup

The case files for the current session are available in this [link](#). Download and extract these files into your run directory. A general overview of the setup is explained below:

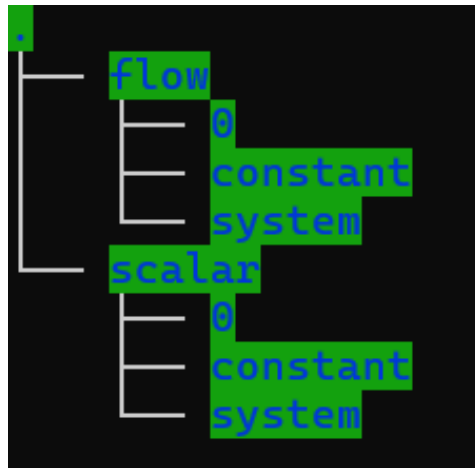


Fig 3 Tree diagram of the main folder

The main folder consists of flow and scalar folder as in Fig 3. The flow folder consists of files to solve the flow field. The scalar folder has the necessary files to simulate the scalar. Since, newScalarTransportFoam, does not solve for velocity, we need to first simulate the flow field. So, initially, the steady state velocity field is solved using flow folder. This flow field is then used by the scalar to move along the flow through local velocity. This is explained more in section 1.6. The folders can be further expanded along this tree:

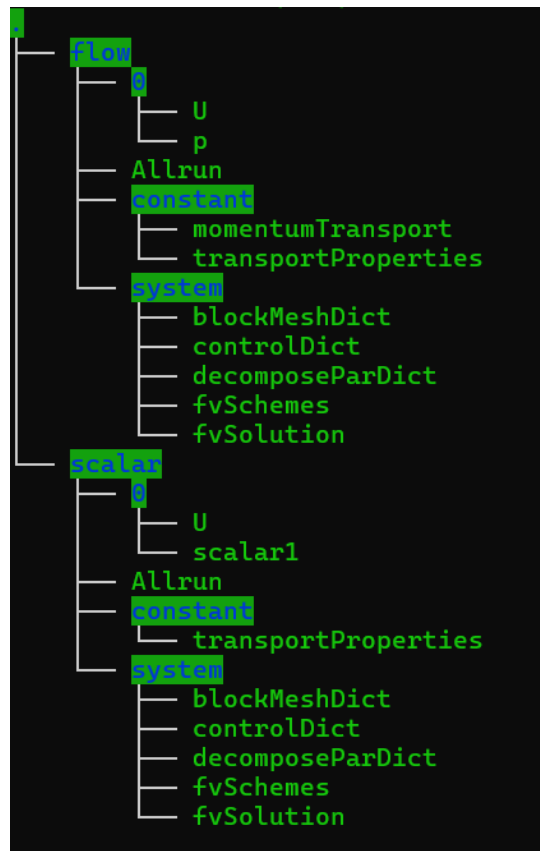


Fig 4 Detailed Tree diagram of the directories

Flow case setup:

The initial and boundary conditions for velocity and pressure are provided in the U and p files of 0 directory. You can see the boundary conditions by accessing these files. The boundary conditions are further explained in the next section.

- The kinematic viscosity of fluid is provided in **constant/transportProperties**.

```

transportModel  Newtonian;

nu              [0 2 -1 0 0 0 0] 8.58E-07;

```

- Similarly, in momentumTransport dictionary, type of model for the simulation is provided. In our case, we have used the laminar model.

```

FoamFile
{
    format      ascii;
    class       dictionary;
    location     "constant";
    object       momentumTransport;
}

simulationType laminar;

```

- The blockMeshDict consists of mesh information and the controlDict dictionary consists of case controls like timing, write information etc.
- System/decomposeParDict dictionary is used for parallel computing.

The steps for the simulation are provided below:

1. First, you need to navigate to the flow folder in your run directory.

```
cd $FOAM_RUN
```

```
cd HalfBore_D_New/flow
```

2. The Allrun file consists of necessary commands to run the simulation. Type **./Allrun** and press enter.

The Allrun file consists of following commands:

```
blockMesh
decomposePar
mpirun -np 6 simpleFoam -parallel
reconstructPar
```

The blockMesh command is used to generate the mesh. Command decomposePar decomposes the domain into subdomains and assigns the number of processors to these subdomains based on the method like simple, scotch etc. In this case, 6 processors are used in parallel and simpleFoam solver is used. At last, reconstructPar command is used to reconstruct a single domain from the processor sub-domains.

Scalar Case setup:

Solver modification and compilation

To simulate the scalar, we have modified the scalarTransportFoam solver and named it newScalarTransportFoam. To make executables for this solver we will need to compile it first. To do that follow the steps given below:

1. Open your terminal and navigate to the run directory.
cd \$FOAM_RUN
2. Navigate to the solver folder by typing the following command.
cd newScalarTransportFoam
3. Compile the solver by typing the following command and press enter.
wclean
wmake

After this the following steps are required.

1. First, you need to navigate to the flow folder.

```
cd HalfBore_D_New /scalar
```

2. The Allrun file consist of necessary commands to run the simulation. Type **./Allrun** and press enter.

The Allrun file consists of following commands:

```
blockMesh
decomposePar
mpirun -np 6 newScalarTransportFoam -parallel
reconstructPar
```

These commands are explained in the previous section.

Transport Properties:

```
FoamFile
{
    format      ascii;
    class        dictionary;
    location     "constant";
    object       transportProperties;
}

DS1              [0 2 -1 0 0 0 0] 1e-6;
kS1              [0 0 -1 0 0 0 0] 0;
```

Here, DS1 is the diffusivity and kS1 is the kinetic rate coefficient. Here, diffusivity is set to 10^{-6} and kinetic rate coefficient to zero since there is no reaction.

1.4.3 Initial and Boundary Conditions

The initial and boundary conditions for the flow and scalar simulations are used separately. The scalar term represents the concentration of the passive scalar in the simulation. At first, the flow is simulated as steady state to obtain a velocity field throughout the cylindrical tube. This flow field is then used for the scalar simulation and a new solver is implemented to simulate the concentration of passive scalar. A steady state solver simpleFoam is used for the flow simulation, whereas an incompressible transient solver newScalarTransportFoam, is used for the scalar transport.

1.4.3.1 Flow Field

The initial conditions are set to zero and the necessary boundary conditions for the flow are tabulated below:

Table 1-1 Boundary Conditions for U

Patch	Condition
Inlet	codedFixedValue
Outlet	zeroGradient
Walls	noslip

Table 1-2 Boundary Conditions for p

Patch	Condition
Inlet	zeroGradient
Outlet	fixedValue(uniform 0)
Walls	zeroGradient

Since, the velocity at the inlet is parabolic and the cylinder is 3D, velocity at various locations (x, y) is calculated using a code which can be accessed in U file of 0 folder.


```

inlet
{
    type          codedFixedValue;
    value          uniform (0 0 0);

    name parabolicVelocity;
    code
    #{
        const vectorField& Cf = patch().Cf();

        vectorField& field = *this;
        const scalar R = 0.005;
        const scalar c = 0;
        const scalar Umax = 0.2;

        forAll(Cf, faceI)
        {
            const scalar x = Cf[faceI][0];
            const scalar y = Cf[faceI][1];

            field[faceI] = vector(0,0,Umax*(1-((pow((y-c)/R,2))+(pow((x-c)/R,2)))));
        }
    #};
}

```

The velocity at the inlet is computed using the coded boundary condition.

A steady state simulation is done to calculate the steady velocity profile in the tube which is used as an input to the scalar which is carried out as transient simulation.

1.4.3.2 Scalar

1.4.3.2.1 Initial Conditions

The steady state velocity profile obtained from the flow simulation is used as a local flow field for the scalar to trace the path of flow. The U file of last time step from the flow simulation is kept in the 0 directory of the scalar.

1.4.3.2.2 Boundary conditions

The flow field is used from the flow simulation and the boundary conditions for the scalar is tabulated below:

codedFixedValue for Scalar:

The scalar is injected at the upper half of the inlet using codedFixedValue boundary condition shown in figure below. This codedFixedValue is compiled and integrated in the OpenFOAM solver as a boundary condition. Here, r represents the y coordinate of the inlet face centers and when $r \geq 0$, the faces are assigned with scalar concentration of 1, else 0.

```

inlet
{
    type          codedFixedValue;
    value          uniform 1;

    name halfBore;
    code
    #{
        const vectorField& Cf = patch().Cf();

        scalarField& field = *this;

        forAll(Cf, faceI)
        {
            const scalar r = Cf[faceI][1];
            if (r>=0)
            {
                field[faceI] = 1;
            }

            else
            {
                field[faceI] = 0;
            }
        }
    #};
}

```

Table 1-3 Boundary Conditions for scalar

Patch	Condition
Inlet – upper half	codedfixedValue - 1
Inlet – lower half	codedfixedValue - 0
Outlet	zeroGradient
Walls	zeroGradient

1.4.3.3 Steady State Study of Scalar

The scalar steady case is to see the steady state of the simulation. If one is interested in the final results or when the solution does not change with time and don't want to trace the scalar with time, one can navigate to this folder **cd HalfBore_D_New /scalar_steady** and type **./Allrun** and press enter. This only provides the steady state solution and observe the final state of the scalar.

2. Results and Discussions

The passive scalar (tracer) is injected from the upper cross section of the inlet with concentration 1 and is gradually moves along the tube with time. The velocity profile at the inlet is parabolic so the tracer is also following it at the time of injection. But after some time, it will cover the whole upper half.

Since, the scalar is now diffusive with non-zero value of diffusion coefficient, the scalar diffuses towards the lower half away from the center as shown in Fig 5. Also, it can be seen in Fig 6 that the scalar diffuses in the boundary layer near the walls resulting in a uniform concentration of 1 over the entire upper half area within simulation time of 20 seconds.

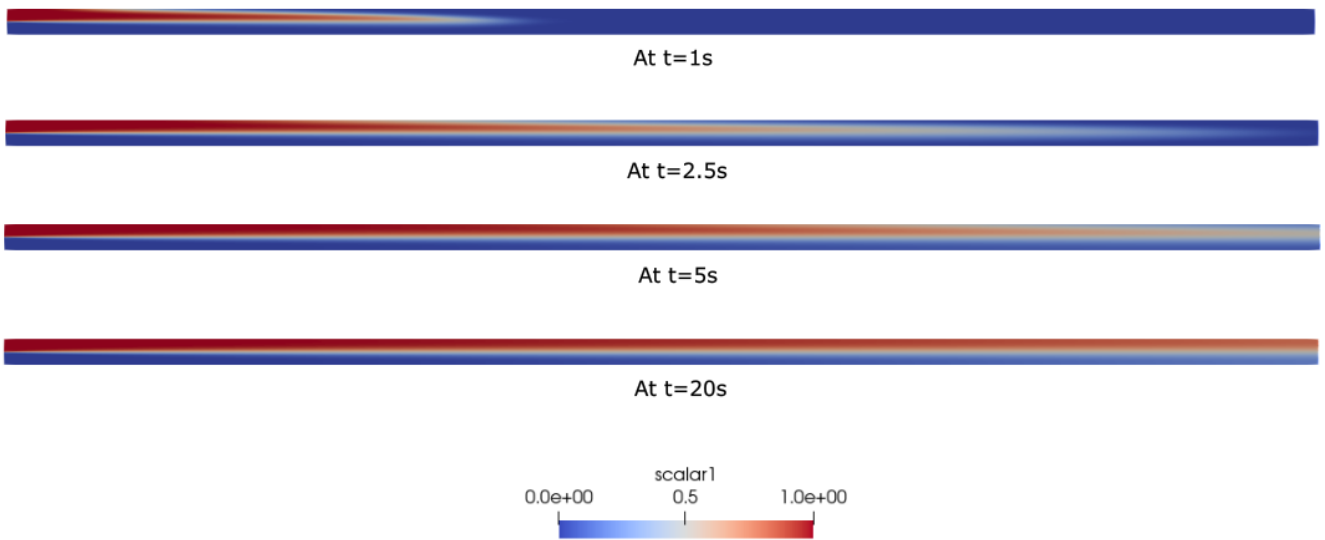


Fig 5 Scalar Concentration along the tube at various time steps

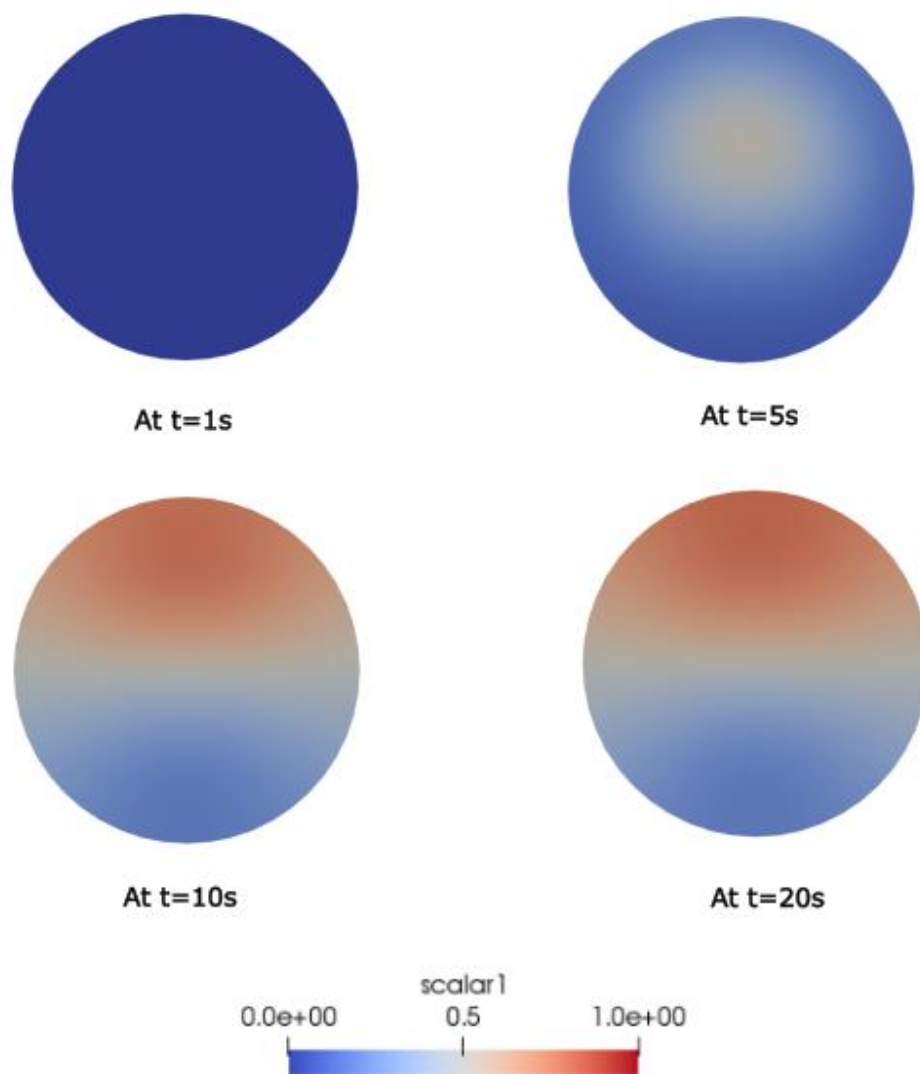


Fig 6 Scalar concentration at the outlet

The cup average concentration is calculated at the outlet and is plotted with time in *Fig 7* to observe the change in concentration at the outlet. The cup average concentration can be calculated as:

$$S_{avg} = \frac{\int u \cdot S \cdot dA}{U_{avg} * A_{outlet}}$$

Where u is the velocity and S is the scalar concentration at area dA . The scalar is averaged over the upper half outlet area to see the effect of diffusion. Due to the movement of tracer in the lower half, the concentration in the upper half is less than without diffusion.

Since the velocity profile is parabolic, the central element will reach the outlet in half the average residence time. Hence the appearance of the tracer occurs at 2.5 seconds where the average residence time is 5 seconds.

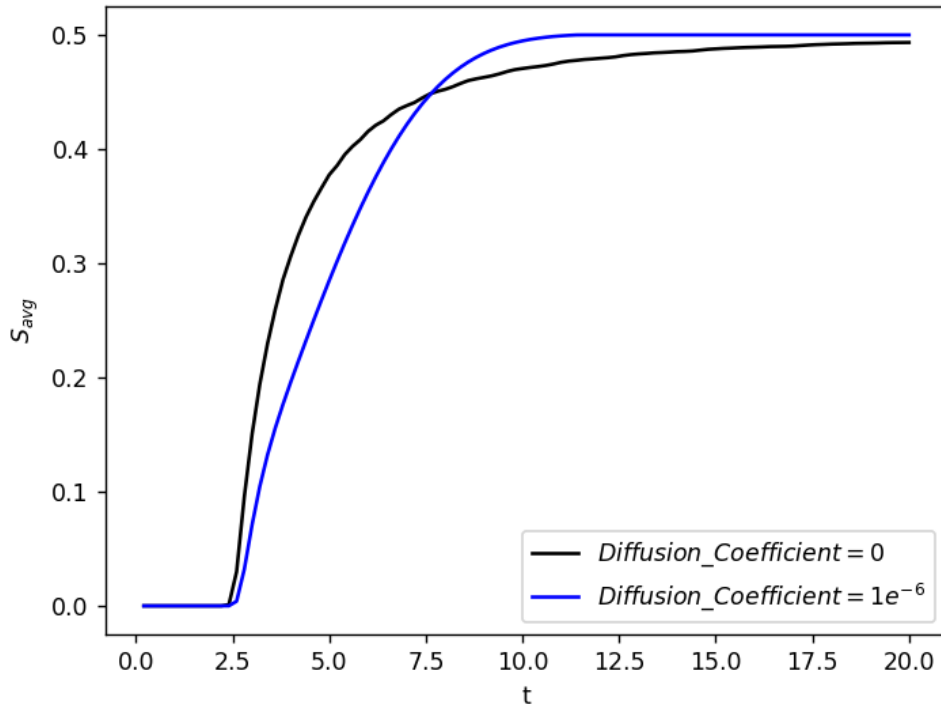


Fig 7 Scalar concentration vs time at the Outlet

The same simulation could be done at different values of diffusion coefficient and different values of velocities to see their impact on tracer distribution along the flow. The same simulation can be done using steady state solver to observe only the final distribution across the radius and length if that is of the interest rather than running the transient solver for long enough to achieve steady state conditions.