



# Calculation of Reaction-Based Residence Time Under Different Operating Conditions and Its Comparison with Flow-Based Residence Time

Dhananjoy Banik Joy<sup>1</sup>, Evan Fernandes<sup>2</sup>, Dr. Dhiraj K. Garg<sup>3</sup>

<sup>1</sup>Department of Mechanical Engineering, Bangladesh University of Engineering and Technology (BUET)

<sup>2</sup>Project Research Assistant, FOSSEE, IIT Bombay

<sup>3</sup>Assistant Professor, Department of Chemical Engineering, Shiv Nadar Institution of Eminence

## Abstract

Residence time, a key factor in the study of reactive fluxes, controls the degree of conversion and the effectiveness of thermal and chemical systems. In this work, the calculation of reaction-based residence time under different operating conditions is investigated and contrasted with traditional flow-based residence time. While flow-based residence time was derived using hydrodynamic and thermal characteristics including reactor geometry and velocity profiles like flat and parabolic inlet, the axial concentration distribution of scalar was determined by solving the scalar transport equation with a first order reaction term. According to the comparison, there are notable variations when heat transfer and non-uniform velocity profiles occur, as reaction kinetics can lengthen or reduce the effective residence period. The results highlight how crucial reaction-based criteria are for accurate reactor and flow performance predictions.

**Keywords:** residence time, concentration, plug flow, parabolic inlet etc.

## Table of Contents

1. Introduction .....	5
2. Problem Statement .....	5
3. Governing Equation .....	
3.1 Governing Equation for flow .....	5
3.2 Governing Equation for scalar .....	6
3.3 Equation for calculating $k_S$ from Arrhenius Equation .....	6
3.4 Equation for reaction-based Residence Time .....	6
4. Simulation Procedure .....	
4.1 Geometry and Meshing .....	7
4.2 Fluid Properties for Case Setup .....	8
4.3 General Case Setup .....	8
4.4 Details Case setup Laminar Isothermal with parabolic Inlet .....	
4.4.1 Flow Setup case .....	10
4.4.2 Initial and Boundary Conditions .....	12
4.4.2.1 Scalar Boundary conditions .....	14
4.4.3 Scalar Properties Setup for Steady and Unsteady Case .....	15
4.5 Details Case setup Laminar Non-Isothermal with parabolic Inlet .....	
4.5.1 Flow Setup case .....	16
4.5.2 Initial and Boundary Conditions.....	17
4.5.2.1 Scalar Boundary conditions.....	19
4.5.3 Scalar Properties Setup .....	20
4.6 Details Case setup Plug Isothermal flow .....	
4.6.1 Flow Setup case .....	21
4.6.2 Initial and Boundary Conditions .....	23
4.6.2.1 Scalar Boundary conditions.....	24
4.6.3 Scalar Properties Setup .....	25
4.7 Details Case setup Plug Non-Isothermal flow .....	
4.7.1 Flow Setup case .....	26
4.7.2 Initial and Boundary Conditions .....	28
4.7.2.1 Scalar Boundary conditions .....	30
4.7.3 Scalar Properties Setup .....	30
4.8 Solver Setup .....	32
5. Results and Discussion .....	
5.1 Flow Based Residence Time.....	33
5.2 Laminar Isothermal Case .....	33
5.3 Laminar Non-Isothermal Case .....	37
5.4 Plug Flow Isothermal Case .....	39
5.5 Plug Flow Non-Isothermal Case .....	41
6. Conclusion .....	43
References .....	43

## List of Figures

Fig 1: Schematic Diagram of the Geometry .....	7
Fig 2: Mesh generation view using blockMesh .....	8
Fig 3: Tree Diagram for Isothermal Laminar Case .....	9
Fig 4: Tree Diagram for Flow Case (flow temperature) .....	9
Fig 5: Tree Diagram for scalar Case .....	10
Fig 6-Fig 46: Flow and scalar case setup	
Fig 47: Solver Setup.....	32
Fig 48: Laminar Flow with velocity profile.....	33
Fig 49: Plug Flow with velocity profile .....	33
Fig 50: Scalar concentration vs time at the Outlet .....	34
Fig 51: Scalar Concentration along the tube (for steady and unsteady) .....	35
Fig 52: Scalar Concentration along various sections of the tube .....	36
Fig 53: Scalar Concentration along various sections of the tube .....	36
Fig 54: Scalar Concentration along the tube .....	37
Fig 55: Scalar Concentration along various sections of the tube .....	38
Fig 56: Scalar Concentration along the tube .....	39
Fig 57: Scalar Concentration along various sections of the tube .....	40
Fig 58: Scalar Concentration along the tube .....	41
Fig 59: Scalar Concentration along various sections of the tube .....	42

## List of Tables

Table 1: Boundary conditions for Laminar Isothermal Case .....	12
Table 2: Boundary conditions for Laminar Non-Isothermal case .....	18
Table 3: Boundary conditions for plug Isothermal case .....	23
Table 4: Boundary conditions for plug non-isothermal case .....	28

## 1. Introduction

The study conveys analysis about reaction-based residence time for 1<sup>st</sup> order kinetic reaction in different operating conditions. The analysis begins with the governing equations of continuity, momentum, and scalar transport to establish the velocity and concentration distributions in laminar pipe flow. By applying this framework under **different operating conditions**—such as plug versus parabolic inlet profiles, and thermal boundary conditions like constant wall heat flux or isothermal conditions—the project explores how reaction kinetics interact with hydrodynamics. For both of the cases no diffusion was taken into consideration. In particular, the study highlights cases where flow-based estimates are insufficient to capture the effective time scales of chemical conversion.

For the design and optimization of reactors, heat exchangers, and pipeline systems—where chemical reactions and transport phenomena are closely related—this comparison is not only theoretically interesting but also practically significant. In laminar parabolic flow, for example, the temperature rise brought on by wall heating speeds up kinetics and modifies the observed conversion, whereas near-wall regions with slower velocity prolong the effective reaction time.

## 2. Problem Statement

Calculation of reaction-based residence time under different operating conditions and its comparison with flow-based residence time.

## 3. Governing Equation

### 3.1 Governing Equation for flow

Continuity Equation for incompressible flow

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

The **Navier–Stokes momentum balance** for a Newtonian fluid is given by

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

where  $\rho$  is the density,  $\mathbf{u}$  is the velocity vector and  $p$  is the pressure and  $\mu$  is the dynamic viscosity.

Solving with symmetry and no-slip at the wall yields the parabolic profile:

$$u_z(r) = 2 u_{\text{avg}}(1 - r^2/R^2) \quad (3)$$

For flat profile,

$$u_{\max} = u_{\text{avg}} \quad (4)$$

From this velocity field, the **flow-based residence time** follows as

$$\tau_{\text{flow}} = L / u_{\text{avg}} \quad (5)$$

### Energy Equation (Thermal Transport)

$$\partial T / \partial t + (\vec{u} \cdot \nabla) T = \alpha \nabla^2 T \quad (6)$$

Where,  $T$ =Temperature,  $\vec{u}$ =Velocity vector,  $\alpha = k / (\rho c_p)$ : thermal diffusivity,  $k$ = thermal conductivity  $C_p$ =specific heat at constant pressure

### 3.2 Governing Equation for Scalar

To describe a reacting species with concentration  $C$ , the scalar transport equation (also called the convection–diffusion–reaction equation) is used (Bird et al., 2002; Levenspiel, 1999):

$$\partial C / \partial t + \vec{u} \cdot \nabla C = D \nabla^2 C - R(C) \quad (7)$$

where  $D$  is the molecular diffusivity and  $R(C)$  is the reaction source term. In our case diffusivity is zero, so  $D = 0$

For a **first-order irreversible reaction** (from  $C \rightarrow D$ )

$$R(C) = k_S C \quad (8)$$

So, the equation will be,

$$\partial C / \partial t + \vec{u} \cdot \nabla C = -k_S C \quad (9)$$

### 3.3 Equation for calculating $k_S$ from Arrhenius Equation

From Arrhenius Equation, we know

$$k_S = A e^{-E_a / RT} \quad (10)$$

Where,  $A$ = pre-exponential factor ( $s^{-1}$ ),  $E_a$ = Activation Energy (J/mol) and  $T$ = Local temperature (K),  $R$ =Universal gas constant(J/mol-K)

### 3.4 Equation for reaction-based Residence Time

$$u_{\text{avg}} dC / dz = -k_S C \quad (11)$$

by integrating, we get

$$C(Z) = C_{in} e^{-(kS/u_{avg}Z)} \quad (12)$$

At the outlet( $Z=L$ )

$$C_{out}=C_{in} e^{-(kS \tau_{flow})} \quad (13)$$

From this, reaction-based residence time is defined as

$$\tau_{reaction} = 1/kS \ln(C_{in}/C_{out}) \quad (14)$$

## 4. Simulation Procedure

### 4.1 Geometry and Meshing

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 chemical species scalar1 will be injected at the inlet and product scalar2 is zero at inlet. As the reaction starts, scalar1 will be converted into scalar2.



Fig 01: Schematic Diagram of the Geometry

The meshing is done using **blockMesh**, openFOAM's built-in tool. The geometry is divided into 5 blocks and 18 vertices. The number of cells is 672000. For complete understanding of Meshing [spoken tutorial](#) can be used.



Fig 2: Mesh generation view using blockMesh

## 4.2 Fluid Properties for Case Setup

Laminar flow with a  $Re=1165$ , is used for flow setup whereas average velocity is  $0.1\text{m/s}$  for parabolic inlet. The kinematic viscosity of water at room temperature ( $300\text{ K}$ ) is  $8.58 \times 10^{-7} \text{ m}^2/\text{s}$ . For Plug flow, average velocity is on the Z direction and the value is  $0.1\text{m/s}$ . For Laminar with parabolic inlet, velocity will be maximum at center and wall will be at no slip condition.  $U_{\text{max}} = 2 \cdot U_{\text{avg}}$ , i.e.  $U_{\text{max}} = 0.2 \text{ m/s}$  for this case. But for plug flow, velocity is constant in the flow direction and so  $U_{\text{max}} = U_{\text{avg}}$  for this case. No diffusivity is taken into consideration for any of the cases. So, Scalar Diffusivity constant will be 0 for all cases. As all of the cases work on  $kS$  as a function of  $T$ , so the values to solve the Arrhenius Equation must be given on the constant/transportProperties while solving for scalar.

## 4.3 General Case Setup

For each case, there are command tree for flow and scalar steady. For both flow and scalar steady case, there are 0, constant and system files. Flow case simulation is run firstly and then the values for the last timestep is used as initial condition for main scalar simulation. For laminar isothermal flow case, both steady and unsteady are observed but for the other cases only steady cases are observed.

For Isothermal laminar case both steady and unsteady cases are observed. We can see the main tree diagram for isothermal laminar case in Fig 3. For the other cases, there will be flow and steady files in the tree.

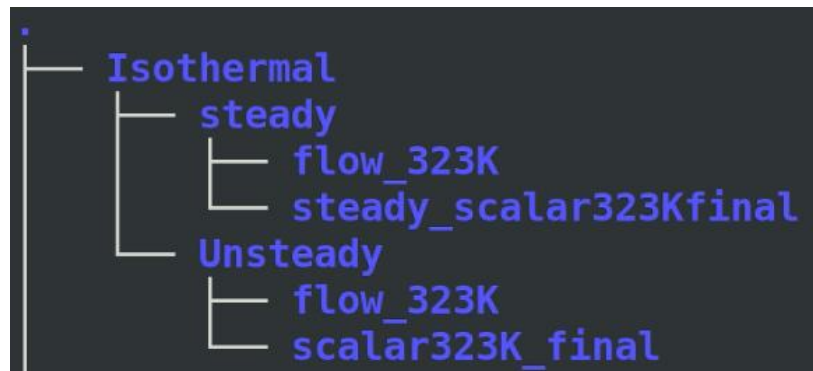


Fig 3: Tree Diagram for Isothermal Laminar Case



In steady case, initially velocity and pressure are solved using **simpleFoam** and then the U and p from the last timestep is used for temperature solve. The wall is kept at constant temperature for both steady and unsteady cases. For each case, there will be flow files for both U, p and T separately.

All flow conditions are taken for water. In the constant folder, transportProperties of water is used. Any flow case for this project is set to be like below:

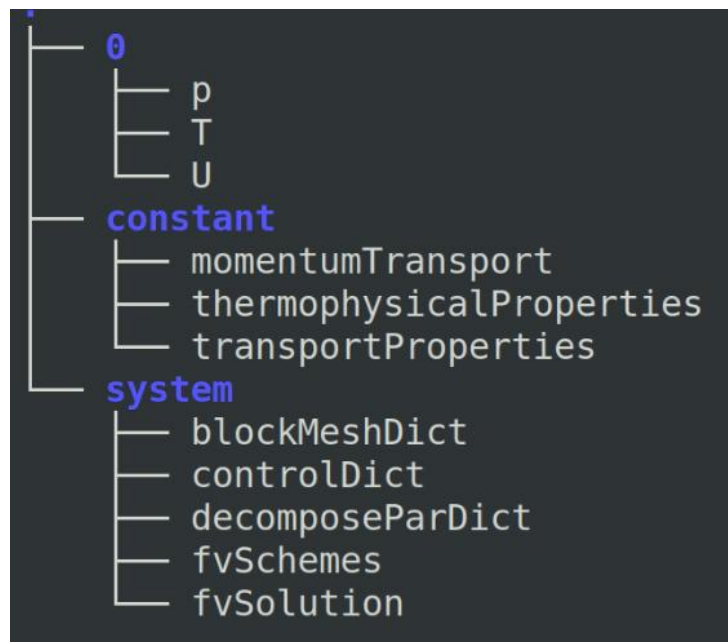


Fig 4: Tree Diagram for Flow Case (flow temperature)

Here, **scalarTransportFoam** solver is used to solve for temperature. This flow case file is similar for unsteady case also. For the non-isothermal case, flow cases will not differ but the initial conditions will be different. Here, in constant/thermophysicalProperties thermodynamic properties for water is defined.

For scalar part, there are initial conditions for scalar1 and scalar2 at  $t=0$ . Solved U and T is used in the 0 directory. A custom solver called “**newScalarTransportFoam1Smodified**” was used to solve. Any scalar (either steady or unsteady) will have the following tree:

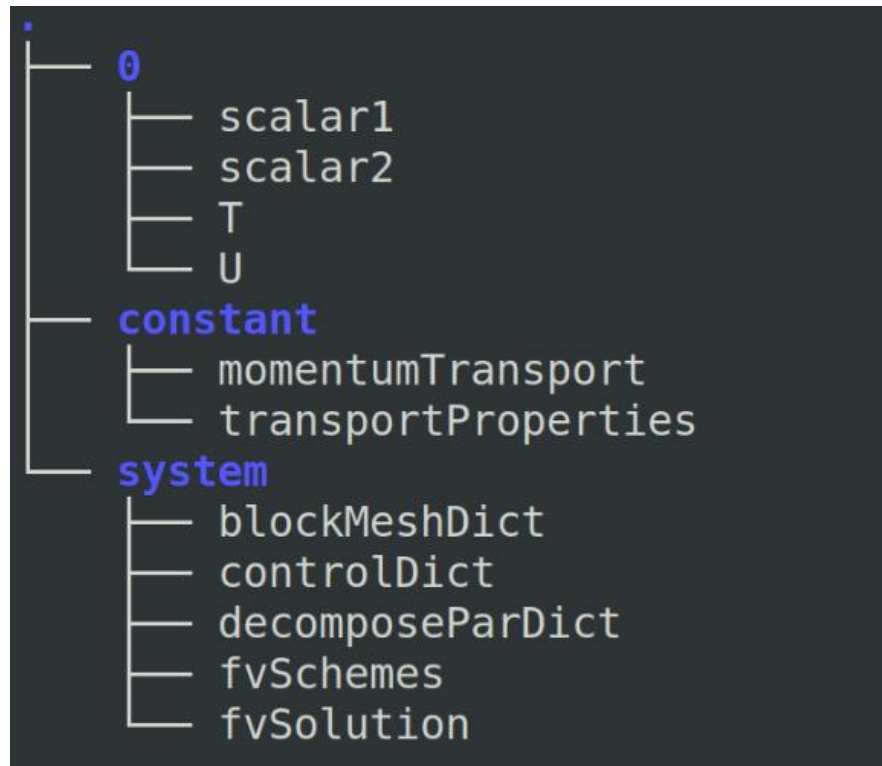


Fig 5: Tree Diagram for scalar Case

In 0, there are conditions for scalar1, scalar2 and also the solved T, U. In the constant folder there are transport values like viscosity and necessary values to calculate  $k_S$  for the local temperature. In momentumTransport simulation type is defined. In the system folder, controlDict defined the start and end timesteps and application for the simulation; decomposeParDict is used to simulate the case in individual processor on a parallel mood which can be reconstructed at the end.

#### 4.4 Details Case setup Laminar Isothermal with parabolic Inlet

##### 4.4.1 Flow Setup case

With all the necessary data and conditions in 0 directory for p and U, case is set. U is given as a codedFixedValue and p is 0 at out.

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

```

DT          DT [0 2 -1 0 0 0 0] 1.43e-7;    // Thermal diffusivity (m^2/s)
nu          nu [0 2 -1 0 0 0 0] 8.58e-7;    // Kinematic viscosity (m^2/s)
transportModel Newtonian;
  
```

Fig 6: Definition of thermal diffusivity and kinetic viscosity in transportProperties file

In the momentumTransport, simulation type is defined. As our case is laminar, we used laminar as simulationType.

```
FoamFile
{
    format      ascii;
    class       dictionary;
    location    "constant";
    object      momentumTransport;
}
// *****
simulationType laminar;
```

Fig 7: Definition of momentumTransport file

For simulation, firstly we have to open OpenFOAM and the navigate to case folder. All the commands for run are given below for solving the velocity:

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

Fig 8: All Run for flow simulation (velocity)

After that, we have to change the application in system/controlDict to run for scalarTransportFoam. As scalarTransportFoam doesn't solve for velocity, we have to use U of the last timestep in the 0/U for the temperature case. All the commands to solve for T are given below:

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

Fig 9: All Run for flow simulation for T

This will solve for T and the T from the last timestep in this case should be used at 0/T for the scalar steady case.

#### 4.4.2 Initial and Boundary Conditions

Separate initial and boundary conditions are used for flow and scalar part. To create a velocity field that scalars can travel through, the flow is first simulated as a steady state.

After that, isothermal conditions are provided on heated walls and isothermal cases are solved for temperature. A custom solver is implemented to model the concentration of passivescalars, and the flow simulation provides the flow field conditions for the scalar simulation.

Boundary conditions for U		Boundary conditions for p		Boundary conditions for T	
Patch	Condition	Patch	Condition	Patch	Condition
Inlet	codedFixedValue	Inlet	zeroGradient	Inlet	fixedValue
Outlet	zeroGradient	Outlet	fixedValue(uniform 0)	Outlet	zeroGradient
Walls	noslip	Walls	zeroGradient	Walls	fixedValue

Table 1: Boundary conditions for Laminar Isothermal Case

Velocity is parabolic at Inlet, so velocity at different locations inside the tube is calculated by using codes which is set in 0/U under flow case.

```

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*(1.0 - ( pow((y - c)/R, 2) + pow((x - c)/R, 2) ))
            );
        }
    #};
}
outlet
{
    type    zeroGradient;
}
wall
{
    type    fixedValue;
    value   uniform (0 0 0);
}

```

Fig 10: Initial Boundary Conditions for parabolic velocity at inlet

We already know that we have to solve for  $T$  in flow. The walls of tube are kept at 323K(50°C) but initial temperature is at 300K (unsteady case), for the steady case, wall temperature is same as initial temperature and inlet.

```

// 0/T
internalField uniform 300; // Initial temperature [K]
boundaryField
{
    inlet
    {
        type      fixedValue;
        value      uniform 323; // Inlet temperature [K]
    }
    outlet
    {
        type      zeroGradient; // Thermally developed flow
    }
    wall
    {
        type      fixedValue; // Isothermal walls
        value      uniform 323; // Wall temperature [K]
    }
}

```

Fig 11: Boundary Conditions for Temperature

#### 4.4.2.1 Scalar Boundary conditions

To find out the scalar concentration at the outlet after running the simulation, both the reactant and product must be given values at  $t=0$ . Scalar1 is used as reactant and scalar2 as product. At inlet, scalar1 is 1 for the full-bore and scalar2 is 0. As the time goes, scalar1 is converted into scalar2 and the concentration for scalar1 decreases which follows 1<sup>st</sup> order reaction formula.

For scalar1 and scalar2 initial conditions are given below:

```

inlet
{
    type      fixedValue;
    value      uniform 1; // inlet = 1 (full bore)
}
outlet
{
    type      zeroGradient;
    value      uniform 0; // initial value = 0 (not enforced later)
}
wall
{
    type      zeroGradient;
    value      uniform 0; // initial value = 0 (not enforced later)
}

```

Fig 12: Scalar Boundary Conditions for scalar 1

```

inlet
{
    type    fixedValue;
    value    uniform 0;    // no product at the inlet
}
outlet
{
    type    zeroGradient; // convective outflow for product
    value    uniform 0;    // initial field on this patch (not enforced later)
}
wall
{
    type    zeroGradient; // no-flux product at walls
    value    uniform 0;    // initial value
}

```

Fig 13: Scalar Boundary Conditions for scalar 2

#### 4.4.3 Scalar Properties Setup for Steady and Unsteady Case

To solve for the scalar case, we have developed a modified scalarTransportFoam solver named newScalarTransportFoam1Smodified which will be discussed later in this report. For unsteady case, we used 6 processors and for steady case we used 8 processors.

```

blockMesh
decomposePar
mpirun -np 6 newScalarTransportFoam1Smodified -parallel

blockMesh
decomposePar
mpirun -np 8 newScalarTransportFoam1Smodified -parallel
reconstructPar

```

Fig 14: All Run for Unsteady and Steady Cases respectively

In the constant folder transportProperties are defined as below:

```

DS1      [0 2 -1 0 0 0 0] 0;    // No diffusion
DS2      [0 2 -1 0 0 0 0] 0;    // No diffusion
DT        DT [0 2 -1 0 0 0 0] 1.43e-7; // Thermal diffusivity (m^2/s)
nu        nu [0 2 -1 0 0 0 0] 8.58e-7; // Kinematic viscosity (m^2/s)
rho       1000;                // Density [kg/m^3]
Cp        4182;                // Specific heat [J/kg-K]
kappa     0.6;                 // Thermal conductivity [W/m-K]
kS
{
    type    Arrhenius;
    A       [0 0 -1 0 0 0 0] 1e6; // Frequency factor (s^-1)
    Ea      [1 2 -2 0 -1 0 0] 50000; // Activation energy (J/mol)
    R       [1 2 -2 -1 -1 0 0] 8.314; // Gas constant (J/mol-K)
}

```

Fig 15: Definition of transport properties for scalar case

Here, we can see that Frequency Factor, Activation Energy and Gas Constant are defined to calculate  $k_S$  using the local temperature which is obtained after solving for temperature. In momentumTransport, same condition as flow is used:

```
FoamFile
{
    format      ascii;
    class       dictionary;
    location    "constant";
    object      momentumTransport;
}
// *****
simulationType laminar;
```

Fig 16: Definition of momentum transport for scalar

In the system/controlDict, timesteps for both unsteady and steady case are given. In the fvSchemes, different schemes for numerical analysis are given and convergence criteria' is given at fvSolution under system folder.

#### 4.5 Details Case setup Laminar Non-Isothermal with parabolic Inlet

##### 4.5.1 Flow Setup case

With all the necessary data and conditions in 0 directory for  $p$  and  $U$ , case is set.  $U$  is given as a codedFixedValue and  $p$  is 0 at out. The kinematic viscosity of fluid is provided in constant/transportProperties.

```
DT          DT [0 2 -1 0 0 0 0] 1.43e-7; // Thermal diffusivity (m^2/s)
nu          nu [0 2 -1 0 0 0 0] 8.58e-7; // Kinematic viscosity (m^2/s)
transportModel Newtonian;
```

Fig 17: Definition of thermal diffusivity and kinematic viscosity in transportProperties file

In the momentumTransport, simulation type is defined. As our case is laminar, we used laminar as simulationType.



```

FoamFile
{
    format      ascii;
    class       dictionary;
    location    "constant";
    object      momentumTransport;
}
// *****
simulationType laminar;

```

Fig 18: Definition of momentumTransport file

For simulation, firstly we have to open OpenFOAM and the navigate to case folder. All the commands for run are given below for solving the velocity:

```

blockMesh
decomposePar
mpirun -np 6 simpleFoam -parallel
reconstructPar

```

Fig 19: All Run for flow simulation (velocity)

After that, we have to change the application in system/controlDict to run for scalarTransportFoam. As scalarTransportFoam doesn't solve for velocity, we have to use U of the last timestep in the 0/U for the temperature case. All the commands to solve for T are given below:

```

blockMesh
decomposePar
mpirun -np 6 scalarTransportFoam -parallel
reconstructPar

```

Fig 20: All Run for flow simulation for T

This will solve for T and the T from the last timestep in this case should be used at 0/T for the scalar steady case.

#### 4.5.2 Initial and Boundary Conditions

Separate initial and boundary conditions are used for flow and scalar part. To create a velocity field that scalars can travel through, the flow is first simulated as a st

eady state. After that, constant heat flux conditions are provided on heated walls and temperature is solved using scalarTransportFoam. A custom solver is implemented to model the concentration of passive scalars, and the flow simulation provides the flow field conditions for the scalar simulation.

Boundary conditions for U		Boundary conditions for p		Boundary conditions for T	
Patch	Condition	Patch	Condition	Patch	Condition
Inlet	fixedValue	Inlet	zeroGradient	Inlet	zeroGradient
Outlet	zeroGradient	Outlet	fixedValue(uniform 0)	Outlet	zeroGradient
Walls	fixedValue	Walls	zeroGradient	Walls	fixedGradient

Table 2: Boundary conditions for Laminar Non-Isothermal case

Velocity is parabolic at Inlet, so velocity at different locations inside the tube is calculated by using codes which is set in 0/U under flow case.

```

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*(1.0 - ( pow((y - c)/R, 2) + pow((x - c)/R, 2) ))
            );
        }
    #};
}
outlet
{
    type    zeroGradient;
}
wall
{
    type    fixedValue;
    value    uniform (0 0 0);
}

```

Fig 21: Initial Boundary Conditions for parabolic velocity at inlet

Now, we already know that we have to solve for T in flow. The walls of tube are kept at constant heat flux, initial temperature and inlet are same; 323K. As the time starts, fluid from inlet to outlet will be heated and the temperature will rise at outlet.

```

inlet
{
    type    zeroGradient;
    value    323;           // convective inflow
}
outlet
{
    type    zeroGradient;   // free outflow
}
wall
{
    type    fixedGradient;
    gradient    uniform -166.67; // = -q''/k
}

```

Fig 22: Boundary Conditions for Temperature

#### 4.5.2.1 Scalar Boundary conditions

To find out the scalar concentration at the outlet after running the simulation, both the reactant and product must be given values at  $t=0$ . Scalar1 is used as reactant and scalar2 as product. At inlet, scalar1 is 1 for the full-bore and scalar2 is 0. As the time goes, scalar1 is converted into scalar2 and the concentration for scalar1 decreases which follows 1<sup>st</sup> order reaction formula.

For scalar1 and scalar2 initial conditions are given below:

```

inlet
{
    type    fixedValue;
    value    uniform 1;    // inlet = 1 (full bore)
}
outlet
{
    type    zeroGradient;
    value    uniform 0;    // initial value = 0 (not enforced later)
}
wall
{
    type    zeroGradient;
    value    uniform 0;    // initial value = 0 (not enforced later)
}

```

```

inlet
{
    type    fixedValue;
    value    uniform 0;    // no product at the inlet
}
outlet
{
    type    zeroGradient; // convective outflow for product
    value    uniform 0;    // initial field on this patch (not enforced later)
}
wall
{
    type    zeroGradient; // no-flux product at walls
    value    uniform 0;    // initial value
}

```

Fig 23: Scalar Boundary Conditions for scalar 1 and scalar 2 (respectively)

### 4.5.3 Scalar Properties Setup

To solve for the scalar case, we have developed a modified `scalarTransportFoam` solver named `newScalarTransportFoam1Smodified` which will be discussed later in this report.

```

blockMesh
decomposePar
mpirun -np 8 newScalarTransportFoam1Smodified -parallel
reconstructPar

```

Fig 24: All Run for steady case

In the constant folder transportProperties are defined as below:

```
DS1      [0 2 -1 0 0 0 0] 0;      // No diffusion
DS2      [0 2 -1 0 0 0 0] 0;      // No diffusion
DT        DT [0 2 -1 0 0 0 0] 1.43e-7;  // Thermal diffusivity (m^2/s)
nu        nu [0 2 -1 0 0 0 0] 8.58e-7;  // Kinematic viscosity (m^2/s)
rho       1000;                    // Density [kg/m^3]
Cp        4182;                    // Specific heat [J/kg-K]
kappa     0.6;                     // Thermal conductivity [W/m-K]
kS
{
    type    Arrhenius;
    A       [0 0 -1 0 0 0 0] 1e6;      // Frequency factor (s^-1)
    Ea      [1 2 -2 0 -1 0 0] 50000;   // Activation energy (J/mol)
    R       [1 2 -2 -1 -1 0 0] 8.314;   // Gas constant (J/mol·K)
}
```

Fig 25: Definition of transport properties for scalar case

Here, we can see that Frequency Factor, Activation Energy and Gas Constant are defined to calculate kS using the local temperature which is obtained after solving for temperature. In momentumTransport, same condition as flow is used:

```
FoamFile
{
    format      ascii;
    class       dictionary;
    location     "constant";
    object       momentumTransport;
}
// ***** //
simulationType laminar;
```

Fig 26: Definition of momentum transport for scalar

In the system/controlDict, timesteps for both unsteady and steady case are given. In the fvSchemes, different schemes for numerical analysis are given and convergence criteria' is given at fvSolution under system folder.

## 4.6 Details Case setup Plug Isothermal Flow

### 4.6.1 Flow Setup case

With all the necessary data and conditions in 0 directory for p and U, case is set. U is constant along the tube and p is 0 at outlet. Velocity at z direction is 0.1m/s.

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

```
DT          DT [0 2 -1 0 0 0] 1.43e-7;  // Thermal diffusivity (m^2/s)
nu          nu [0 2 -1 0 0 0] 8.58e-7;  // Kinematic viscosity (m^2/s)
transportModel Newtonian;
```

Fig 27: Definition of thermal diffusivity and kinematic viscosity in transportProperties file

In the momentumTransport, simulation type is defined. As our case is laminar, we used laminar as simulationType.

```
FoamFile
{
    format      ascii;
    class       dictionary;
    location    "constant";
    object      momentumTransport;
}
// *****
simulationType laminar;
```

Fig 28: Definition of momentumTransport file

For simulation, firstly we have to open OpenFOAM and the navigate to case folder. All the commands for run are given below for solving the velocity:

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

Fig 29: All Run for flow simulation (velocity)

After that, we have to change the application in system/controlDict to run for scalarTransportFoam. As scalarTransportFoam doesn't solve for velocity, we have to use U of the last timestep in the 0/U for the temperature case. All the commands to solve for T are given below:

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

Fig 30: All Run for flow simulation for T

This will solve for T and the T from the last timestep in this case should be used at 0/T for the scalar steady case.

#### 4.6.2 Initial and Boundary Conditions

Separate initial and boundary conditions are used for flow and scalar part. To create a velocity field that scalars can travel through, the flow is first simulated as a steady state. After that, isothermal conditions are provided on heated walls and temperature is solved using scalarTransportFoam. A custom solver is implemented to model the concentration of passive scalars, and the flow simulation provides the flow field conditions for the scalar simulation.

Boundary conditions for U		Boundary conditions for p		Boundary conditions for T	
Patch	Condition	Patch	Condition	Patch	Condition
Inlet	fixedValue	Inlet	zeroGradient	Inlet	zeroGradient
Outlet	zeroGradient	Outlet	fixedValue(uniform 0)	Outlet	zeroGradient
Walls	fixedValue	Walls	zeroGradient	Walls	fixedValue

Table 3: Boundary conditions for plug Isothermal case

Velocity is flat at Inlet, so value of velocity is given in the initial condition which is set in 0/U under flow case.

```

inlet
{
    type    fixedValue;
    value    uniform (0 0 0.1);
}
outlet
{
    type    zeroGradient;
}
wall
{
    type    fixedValue;
    value    uniform (0 0 0.1);
}

```

Fig 31: Initial Boundary Conditions for Flat Velocity at inlet

We already know that we have to solve for  $T$  in flow. The walls of tube are kept at 323K(50°C), as this is steady case, initial, inlet and walls are at same temperature.

```

inlet
{
    type    fixedValue;
    value    uniform 323;    // Inlet temperature [K]
}
outlet
{
    type    zeroGradient;    // Thermally developed flow
}
wall
{
    type    fixedValue;    // Isothermal walls
    value    uniform 323;    // Wall temperature [K]
}
// Alternative for adiabatic walls:
// type    zeroGradient;

```

Fig 32: Boundary Conditions for Temperature

#### 4.6.2.1 Scalar Boundary conditions

To find out the scalar concentration at the outlet after running the simulation, both the reactant and product must be given values at  $t=0$ . Scalar1 is used as reactant and scalar2 as product. At inlet, scalar1 is 1 for the full-bore and scalar2 is 0. As the time goes,



scalar1 is converted into scalar2 and the concentration for scalar1 decreases which follows 1<sup>st</sup> order reaction formula.

For scalar1 and scalar2 initial conditions are given below:

```

inlet
{
    type    fixedValue;
    value    uniform 1;    // inlet = 1 (full bore)
}
outlet
{
    type    zeroGradient;
    value    uniform 0;    // initial value = 0 (not enforced later)
}
wall
{
    type    zeroGradient;
    value    uniform 0;    // initial value = 0 (not enforced later)
}

```

```

inlet
{
    type    fixedValue;
    value    uniform 0;    // no product at the inlet
}
outlet
{
    type    zeroGradient; // convective outflow for product
    value    uniform 0;    // initial field on this patch (not enforced later)
}
wall
{
    type    zeroGradient; // no-flux product at walls
    value    uniform 0;    // initial value
}

```

Fig 33: Scalar Boundary Conditions for scalar 1 and scalar 2 (respectively)

#### 4.6.3 Scalar Properties Setup

To solve for the scalar case, we have developed a modified scalarTransportFoam solver named newScalarTransportFoam1Smodified which will be discussed later in this report.

```

blockMesh
decomposePar
mpirun -np 8 newScalarTransportFoam1Smodified -parallel
reconstructPar

```

Fig 34: All Run for steady case

In the constant folder transportProperties are defined as below:

```

DS1      [0 2 -1 0 0 0 0] 0;      // No diffusion
DS2      [0 2 -1 0 0 0 0] 0;      // No diffusion
DT       DT [0 2 -1 0 0 0 0] 1.43e-7; // Thermal diffusivity (m^2/s)
nu       nu [0 2 -1 0 0 0 0] 8.58e-7; // Kinematic viscosity (m^2/s)
rho      1000;                      // Density [kg/m^3]
Cp       4182;                      // Specific heat [J/kg-K]
kappa    0.6;                      // Thermal conductivity [W/m-K]
kS
{
    type    Arrhenius;
    A       [0 0 -1 0 0 0 0] 1e6;    // Frequency factor (s^-1)
    Ea      [1 2 -2 0 -1 0 0] 50000; // Activation energy (J/mol)
    R       [1 2 -2 -1 -1 0 0] 8.314; // Gas constant (J/mol-K)
}

```

Fig 35: Definition of transport properties for scalar case

Here, we can see that Frequency Factor, Activation Energy and Gas Constant are defined to calculate  $kS$  using the local temperature which is obtained after solving for temperature. In momentumTransport, same condition as flow is used:

```

FoamFile
{
    format      ascii;
    class       dictionary;
    location    "constant";
    object      momentumTransport;
}
// ***** //
simulationType laminar;

```

Fig 36: Definition of momentum transport for scalar

In the system/controlDict, timesteps for both unsteady and steady case are given. In the fvSchemes, different schemes for numerical analysis are given and convergence criteria' is given at fvSolution under system folder.

## 4.7 Details Case setup for Plug Non-Isothermal Flow

### 4.7.1 Flow Setup case

With all the necessary data and conditions in 0 directory for  $p$  and  $U$ , case is set.  $U$  is constant along the tube and  $p$  is 0 at outlet. Velocity at  $z$  direction is 0.1m/s. The kinematic viscosity of fluid is provided in constant/transportProperties.

```
DT          DT [0 2 -1 0 0 0] 1.43e-7; // Thermal diffusivity (m^2/s)
nu          nu [0 2 -1 0 0 0] 8.58e-7; // Kinematic viscosity (m^2/s)
transportModel Newtonian;
```

Fig 37: Definition of thermal diffusivity and kinematic viscosity in transportProperties file

In the momentumTransport, simulation type is defined. As our case is laminar, we used laminar as simulationType.

```
FoamFile
{
    format      ascii;
    class       dictionary;
    location    "constant";
    object      momentumTransport;
}
// *****
simulationType laminar;
```

Fig 38: Definition of momentumTransport file

For simulation, firstly we have to open OpenFOAM and the navigate to case folder. All the commands for run are given below for solving the velocity:

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

Fig 39: All Run for flow simulation (velocity)

After that, we have to change the application in system/controlDict to run for scalarTransportFoam. As scalarTransportFoam doesn't solve for velocity, we have to use U of the last timestep in the 0/U for the temperature case. All the commands to solve for T are given below:

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

Fig 40: All Run for flow simulation for T

This will solve for T and the T from the last timestep in this case should be used at 0/T for the scalar steady case.

#### 4.7.2 Initial and Boundary Conditions

Separate initial and boundary conditions are used for flow and scalar part. To create a velocity field that scalars can travel through, the flow is first simulated as a steady state. After that, constant heat flux conditions are provided on heated walls and temperature is solved using scalarTransportFoam. A custom solver is implemented to model the concentration of passive scalars, and the flow simulation provides the flow field conditions for the scalar simulation.

Boundary conditions for U		Boundary conditions for p		Boundary conditions for T	
Patch	Condition	Patch	Condition	Patch	Condition
Inlet	fixedValue	Inlet	zeroGradient	Inlet	zeroGradient
Outlet	zeroGradient	Outlet	fixedValue(uniform 0)	Outlet	zeroGradient
Walls	fixedValue	Walls	zeroGradient	Walls	fixedGradient

Table 4: Boundary conditions for plug non-isothermal case

Velocity is flat profile at Inlet and remains constant along the tube length, so velocity 0.1m/s is set in 0/U under flow case.

```

inlet
{
    type    fixedValue;
    value    uniform (0 0 0.1);
}
outlet
{
    type    zeroGradient;
}
wall
{
    type    fixedValue;
    value    uniform (0 0 0.1);
}

```

Fig 41: Initial Boundary Conditions for Flat Velocity at inlet

We already know that we have to solve for  $T$  in flow. The walls of tube are kept at fixed gradient (constant heat flux) and as it is a steady case, initial and inlet temperature is kept as same (323K).

```

inlet
{
    type    zeroGradient;
    value    323;           // convective inflow
}
outlet
{
    type    zeroGradient;   // free outflow
}
wall
{
    type    fixedGradient;
    gradient    uniform -166.67; // = -q''/k
}

```

Fig 42: Boundary Conditions for Temperature

#### 4.7.2.1 Scalar Boundary conditions

To find out the scalar concentration at the outlet after running the simulation, both the reactant and product must be given values at  $t=0$ . Scalar1 is used as reactant and scalar2 as product. At inlet, scalar1 is 1 for the full-bore and scalar2 is 0. As the time goes, scalar1 is converted into scalar2 and the concentration for scalar1 decreases which follows 1<sup>st</sup> order reaction formula.

For scalar1 and scalar2 initial conditions are given below:

```
inlet
{
    type    fixedValue;
    value    uniform 1;    // inlet = 1 (full bore)
}
outlet
{
    type    zeroGradient;
    value    uniform 0;    // initial value = 0 (not enforced later)
}
wall
{
    type    zeroGradient;
    value    uniform 0;    // initial value = 0 (not enforced later)
}
```

```
inlet
{
    type    fixedValue;
    value    uniform 0;    // no product at the inlet
}
outlet
{
    type    zeroGradient; // convective outflow for product
    value    uniform 0;    // initial field on this patch (not enforced later)
}
wall
{
    type    zeroGradient; // no-flux product at walls
    value    uniform 0;    // initial value
}
```

Fig 43: Scalar Boundary Conditions for scalar 1 and scalar 2 (respectively)

#### 4.7.3 Scalar Properties Setup

To solve for the scalar case, we have developed a modified scalarTransportFoam solver named newScalarTransportFoam1Smodified which will be discussed later in this report.

```

blockMesh
decomposePar
mpirun -np 8 newScalarTransportFoam1smodified -parallel
reconstructPar

```

Fig 44: All Run for steady case

In the constant folder transportProperties are defined as below:

```

DS1      [0 2 -1 0 0 0 0] 0;      // No diffusion
DS2      [0 2 -1 0 0 0 0] 0;      // No diffusion
DT       DT [0 2 -1 0 0 0 0] 1.43e-7; // Thermal diffusivity (m^2/s)
nu       nu [0 2 -1 0 0 0 0] 8.58e-7; // Kinematic viscosity (m^2/s)
rho      1000;                      // Density [kg/m^3]
Cp       4182;                      // Specific heat [J/kg-K]
kappa    0.6;                      // Thermal conductivity [W/m-K]
kS
{
    type    Arrhenius;
    A       [0 0 -1 0 0 0 0] 1e6;    // Frequency factor (s^-1)
    Ea      [1 2 -2 0 -1 0 0] 50000; // Activation energy (J/mol)
    R       [1 2 -2 -1 -1 0 0] 8.314; // Gas constant (J/mol·K)
}

```

Fig 45: Definition of transport properties for scalar case

Here, we can see that Frequency Factor, Activation Energy and Gas Constant are defined to calculate kS using the local temperature which is obtained after solving for temperature.

In momentumTransport, same condition as flow is used:

```

FoamFile
{
    format      ascii;
    class       dictionary;
    location    "constant";
    object      momentumTransport;
}
// *****
simulationType laminar;

```

Fig 46: Definition of momentum transport for scalar case

In the system/controlDict, timesteps for both unsteady and steady case are given. In the fvSchemes, different schemes for numerical analysis are given and convergence criteria' is given at fvSolution under system folder.

## 4.8 Solver Setup

As we know, we have used a modified scalarTransportFoam which can be used for 1<sup>st</sup> order kinetic reaction and the reaction rate constant is calculated from the local fluid temperature. Firstly, we have to compile the solver and then we can use the solver for our case.

```
// Update reaction rate based on current temperature
kS = A*exp(-Ea/(R*T));
```

```
fvScalarMatrix scalar1Eqn
(
    fvm::ddt(scalar1)
    + fvm::div(phi, scalar1)
    - fvm::laplacian(DS1, scalar1)
    ==
    -kS*scalar1    // Now temperature-dependent
);
fvScalarMatrix scalar2Eqn
(
    fvm::ddt(scalar2)
    + fvm::div(phi, scalar2)
    - fvm::laplacian(DS2, scalar2)
    ==
    kS*scalar1
);
```

Fig 47: Solver Setup



## 5. Results and Discussion

### 5.1 Flow Based Residence Time

For Laminar Flow, Length of the tube=0.5m and  $u_{\max}=0.2$  m/s at the center

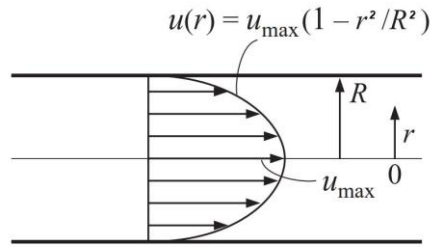


Fig 48: Laminar Flow with velocity profile

So, residence time will be:  $t = L/U_{\text{avg}} = \frac{0.5}{0.1} = 5\text{s}$ ; where at the center  $t = \frac{L}{U_{\max}} = 2.5\text{s}$

For Ideal plug flow with flat velocity,

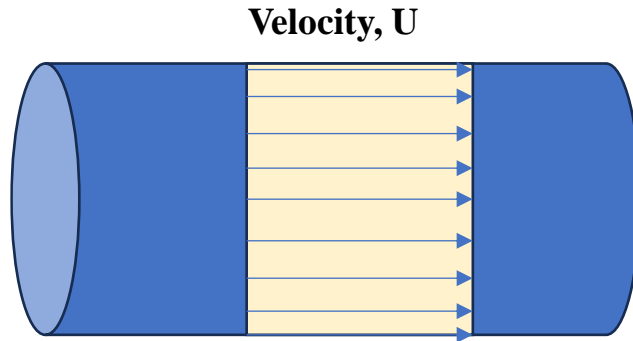


Fig 49: Plug Flow with velocity profile

$U_{\text{avg}} = U_{\max}$  along the flow; residence time,  $t = \frac{0.5}{0.1} = 5\text{s}$ , from wall to center there is no variation for residence time in this case.

### 5.2 Laminar Isothermal Case:

Both steady and unsteady cases are solved. For unsteady, 20 timesteps are taken. From  $t=0$ , when the reaction starts, scalar1 starts to convert into scalar2 along with time. For unsteady case, first appearance of tracer at the outlet happens at 2.5s which is **half** of the flow-based residence time. From **Hagen–Poiseuille law**, we know that at the center the velocity is maximum and is twice the average velocity. Around 6 times the residence time, scalar1

seems to be static. After the flow reaches the steady state average cup concentration is calculated at outlet.

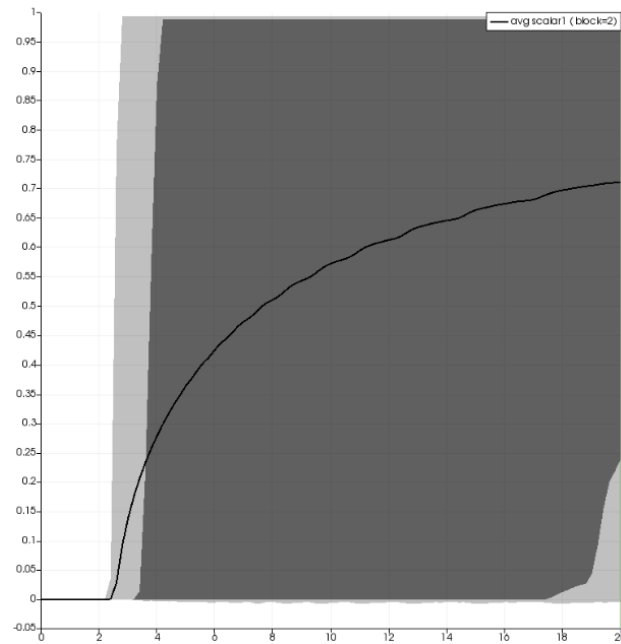
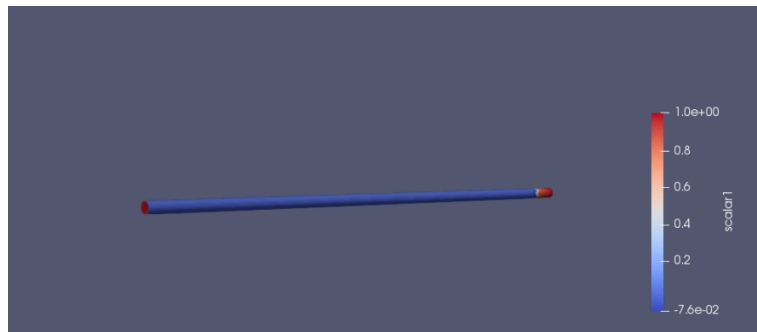


Fig 50: Scalar concentration vs time at the Outlet

For steady case, the simulation took 191s, scalar1 concentration is maximum at inlet at  $t=0$ . As the time goes, conversion happens. As the velocity profile is parabolic, velocity is minimum at wall but conversion is maximum at wall.



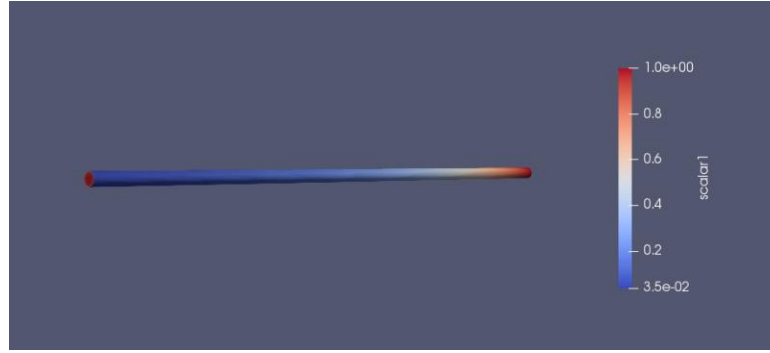


Fig 51: Scalar Concentration along the tube (for steady and unsteady)

### Cup average concentration and keff is calculated at outlet:

At the outlet plane A, the **cup average concentration** is the single number that conserves the total scalar outflow

$$C_{avg} = \int \frac{u \cdot C dA}{U_{avg} \times A_{outlet}}$$

Where U is the velocity and S is the scalar concentration at the X-section dA.

The concentration of scalar1 decreases and scalar2 increases as the fluid flows along the tube which can be seen in Fig 8. As there is a thermal boundary layer inside the tube, we can have the “Flux Weighted Axial Average  $k_{eff}$ ” which is expressed as:

$$K_{eff} = \frac{1}{L} \int_0^L \frac{\int kS(T)C(U \cdot n) dA}{\int (U \cdot n) C dA} dz$$

This equation helps to calculate the kS value over the contour. We can calculate effective residence time for reaction by using 1st order reaction rate formula

Cup average concentration at outlet along the tube length is shown on the following figures both for unsteady and steady:

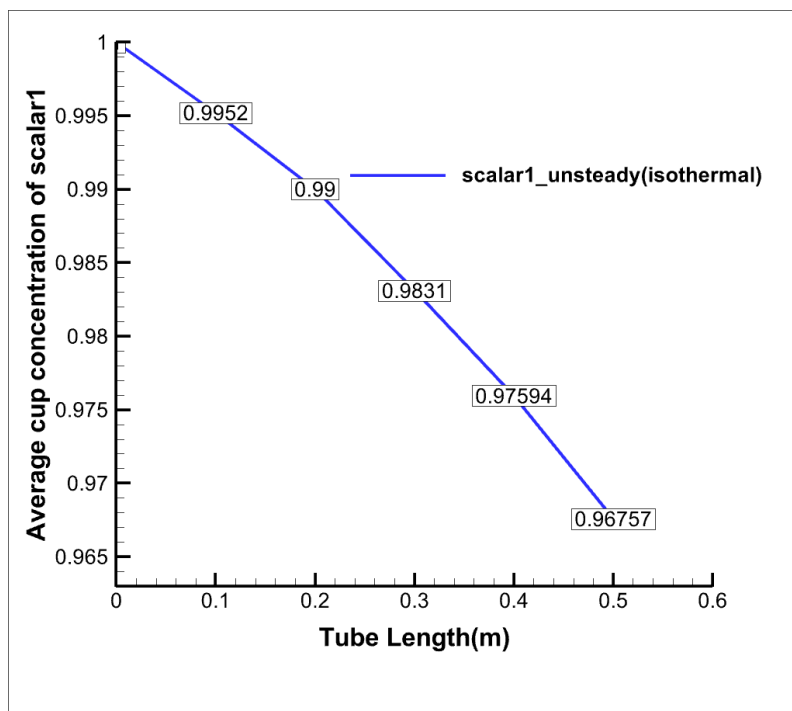


Fig 52: Scalar Concentration along various sections of the tube

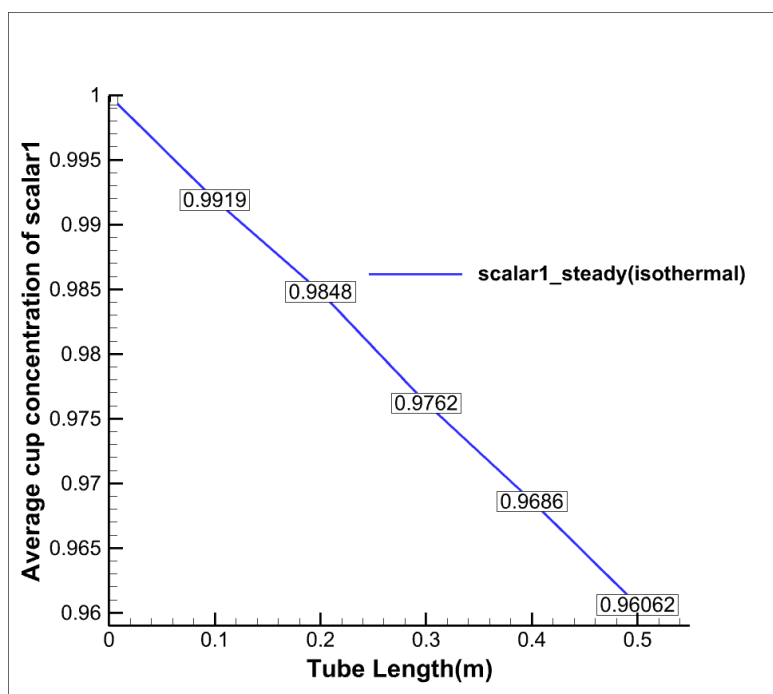


Fig 53: Scalar Concentration along various sections of the tube

From these concentration values, reaction-based residence time can be calculated.

### Reaction based Residence Time Calculation:

For steady case,

To calculate residence time, we have average cup concentration of scalar1 at outlet and flux weighted  $k_{eff}$ . We know for 1<sup>st</sup> order reaction,

$$C_a = C_0 e^{-k_{eff} t} \quad k_{eff} = 0.0082, C_a = 0.96062, C_0 = 1$$

From that, we find,  $t = 4.89s$ , which is almost same as **Flow Based Residence Time**

### 5.3 Laminar Non-Isothermal Case

For this case, scalar1 is injected at full-bore of inlet. The scalar1 gets converted to scalar2 as the flow proceeds. At the initial sections of the tube near the inlet, scalar1 concentration is higher but as the time progresses and flow marches forward, it starts converting to scalar2. Since the velocity is parabolic inlet, the velocity is maximum at center and 0 at wall and conversion will be higher at walls. Slower  $u$  leads to higher  $L/u$ , and more conversion from reactant to product. Scalar concentration is shown in the following figure along the tube:

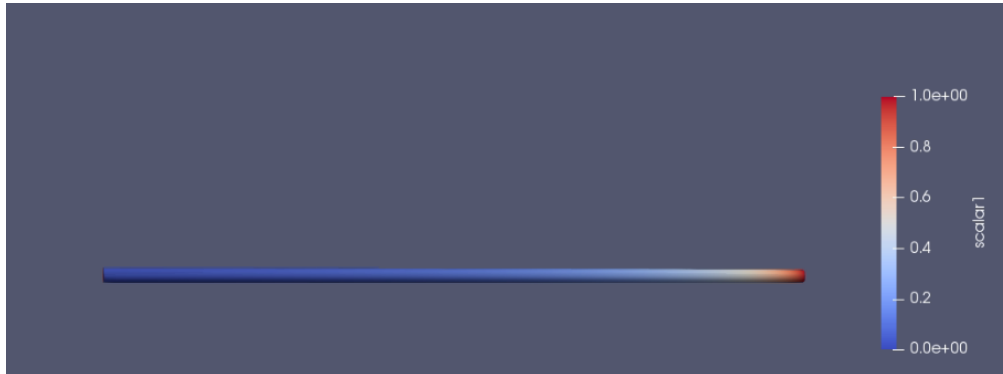


Fig 54: Scalar Concentration along the tube

#### Cup average concentration and $k_{eff}$ is calculated at outlet:

At the outlet plane A, the **cup average concentration** is the single number that conserves the total scalar outflow

$$C_{avg} = \frac{\int U n C dA}{\int U n dA}$$

Where  $U$  is the velocity and  $S$  is the scalar concentration at the  $X$ -section  $dA$ .

The concentration of scalar1 decreases and scalar2 increases as the fluid flows along the tube which can be seen in Fig 8. As there is a thermal boundary layer inside the tube, we can have the “Flux Weighted Axial Average  $k_{eff}$ ” which is expressed as:

$$k_{\text{eff}} = \frac{1}{L} \int_0^L \frac{\int k_S(T) C(U.n) dA}{\int (U.n) C dA} dz$$

This equation helps to calculate the  $k_S$  value over the contour. We can calculate effective residence time for reaction by using 1st order reaction rate formula.

From inlet to outlet along the tube average cup concentration for scalar1 is shown in the figure:

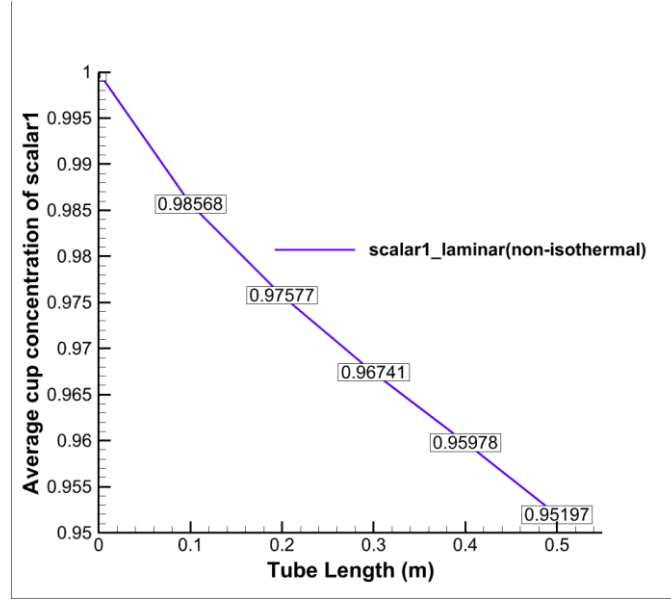


Fig 55: Scalar Concentration along various sections of the tube

Here, from this value, we can calculate the reaction-based residence time for scalar1 under non-isothermal heating conditions.

### Reaction based Residence Time Calculation:

To calculate residence time, we have average cup concentration of scalar1 at outlet and flux weighted  $k_{\text{eff}}$ . We know for 1<sup>st</sup> order reaction,

$$C_a = C_0 e^{-k_{\text{eff}} t} \quad k_{\text{eff}} = 0.00821, C_a = 0.95197, C_0 = 1$$

From that, we find,  $t = 5.99\text{s}$ , which differs slightly from **Flow Based Residence Time**.

### 5.4 Plug flow Isothermal Case

For plug flow isothermal case, initial and inlet is kept at same temperature like wall (323K). As time goes, scalar1 is converted to scalar2 and for t=194s we can see the scalar1 distribution along the tube. For ideal PFR, conversion is same everywhere across the cross section. There's no difference in conversion between wall and center.

$C(r) = 1 - e^{-\frac{kS T_0 L}{U}}$  ; independent of radial position  
We can see the scalar distribution along the tube:

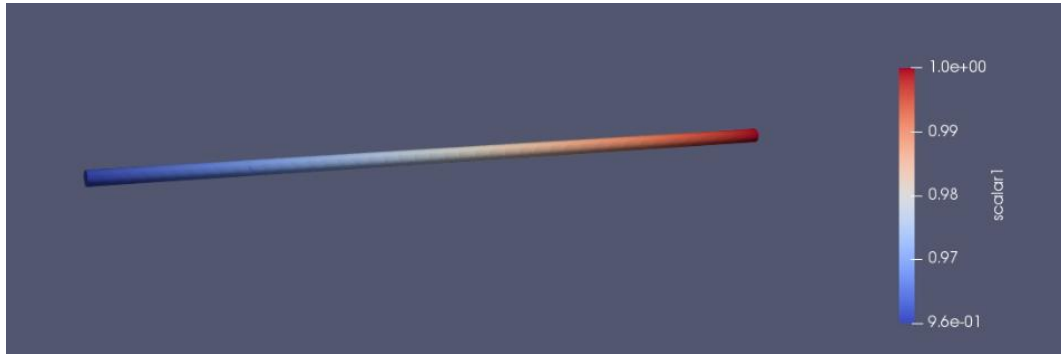


Fig 56: Scalar Concentration along the tube

**Cup average concentration and  $k_{eff}$  is calculated at outlet:**

At the outlet plane A, the **cup average concentration** is the single number that conserves the total scalar outflow

$$C_{avg} = \frac{\int U n C dA}{\int U n dA}$$

Where U is the velocity and S is the scalar concentration at the X-section dA.

The concentration of scalar1 decreases and scalar2 increases as the fluid flows along the tube which can be seen in Fig 8. We can have the “Flux Weighted Axial Average  $k_{eff}$ ” which is expressed as:

$$k_{eff} = \frac{1}{L} \int_0^L \frac{kS(T)C(U.n)dA}{\int (U.n)C dA} dz$$

This equation helps to calculate the kS value over the contour. We can calculate effective residence time for reaction by using 1st order reaction rate formula.

By calculating, we can have the cup average outlet concentration from inlet to outlet along the tube in the following figure:

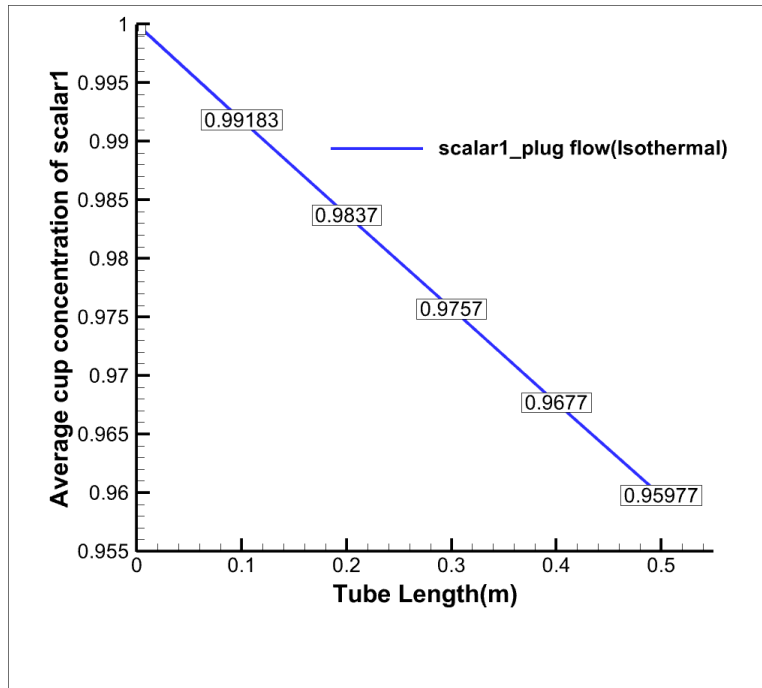


Fig. 57: Scalar Concentration along various sections of the tube

### Reaction based Residence Time Calculation:

To calculate residence time, we have average cup concentration of scalar1 at outlet and flux weighted  $kS$ . We know for 1<sup>st</sup> order reaction,

$$C_a = C_0 e^{-k_{\text{eff}} t} \quad k_{\text{eff}} = 0.0082, C_a = 0.95977, C_0 = 1$$

From that, we find,  $t = 5.0075\text{s}$ , which is almost same as **Flow Based Residence Time**.



### 5.5 Plug Flow Non-Isothermal Case

In this case, there is temperature gradient from wall to flow. Wall is kept at constant heat flux and heating up the flow will cause temperature difference  $T(r,Z)$ . Inlet is kept at 323K and temperature will rise at the outlet. Residence time is identical for all streamlines, so differences come **only from temperature** via  $kS(T)$ :

$$C(r) = 1 - e^{-\frac{1}{U} \int_0^L kS(T(r,Z)) dz}$$

As the flow will be heated in this condition, more conversion is happening near wall due to temperature. From inlet to outlet along the tube scalar1 distribution is shown in the figure:

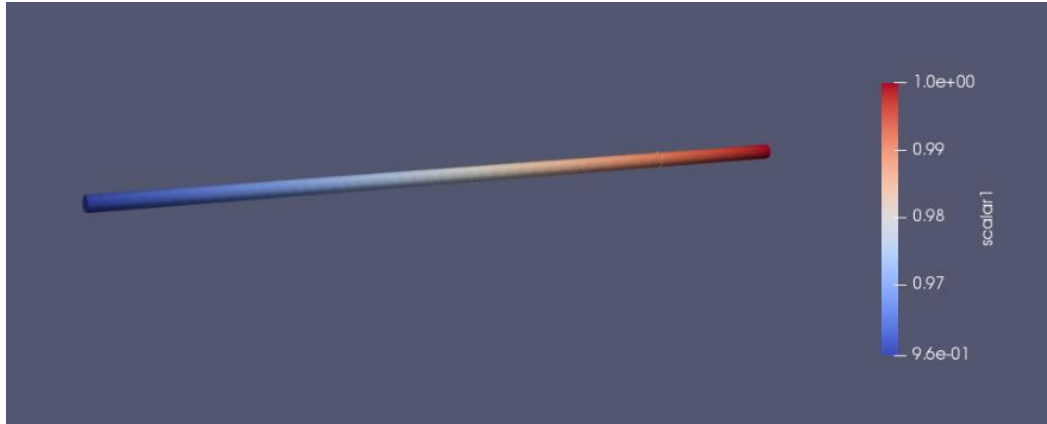


Fig 58: Scalar Concentration along the tube

**Cup average concentration and  $k_{eff}$  is calculated at outlet:**

At the outlet plane A, the **cup average concentration** is the single number that conserves the total scalar outflow

$$C_{avg} = \frac{\int U n C dA}{\int U n dA}$$

Where U is the velocity and S is the scalar concentration at the X-section dA.

The concentration of scalar1 decreases and scalar2 increases as the fluid flows along the tube which can be seen in Fig 8. We can have the “Flux Weighted Axial Average  $k_{eff}$ ” which is expressed as:

$$k_{eff} = \frac{1}{L} \int_0^L \frac{\int kS(T)C(U.n)dA}{\int (U.n)C dA} dz$$

This equation helps to calculate the  $kS$  value over the contour. We can calculate effective residence time for reaction by using 1st order reaction rate formula.

By calculating, we can draw the cup average concentration across different sections of the tube:

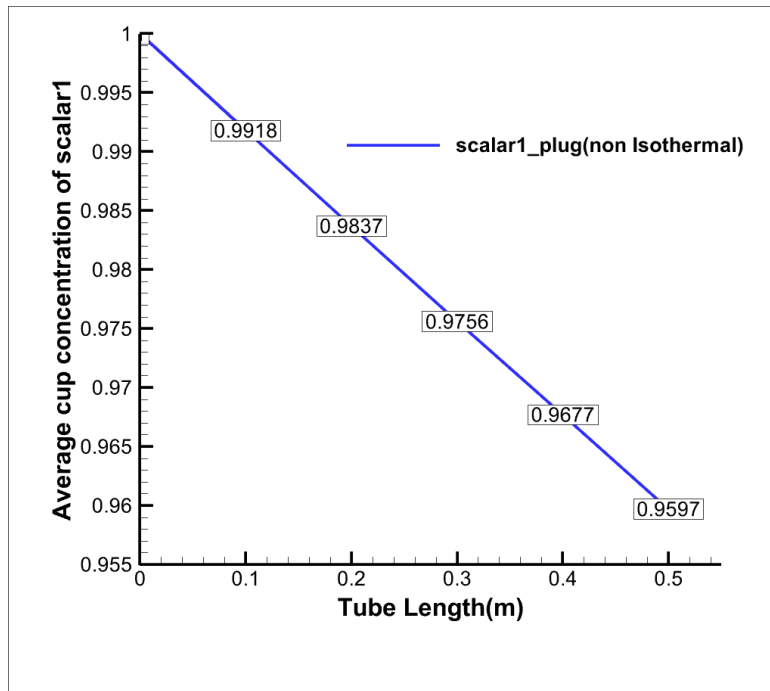


Fig 59: Scalar Concentration along various sections of the tube

Here, from this value, we can calculate the reaction-based residence time for scalar1 under isothermal heating conditions.

### Reaction based Residence Time Calculation:

To calculate residence time, we have average cup concentration of scalar1 at outlet and flux weighted  $k_{\text{eff}}$ . We know for 1<sup>st</sup> order reaction,

$$C_a = C_0 e^{-k_{\text{eff}} t} \quad k_{\text{eff}} = 0.00824, C_a = 0.9597, C_0 = 1$$

From that, we find,  $t = 4.99\text{s}$ , which is almost same as **Flow Based Residence Time**.

## 6 Conclusion

For the two different flow and heating conditions, reaction based and flow-based residence time is observed. For Laminar flow with parabolic inlet where the velocity at center is twice the average velocity of the flow, we found reaction-based residence time of about 4.89s and 5.99s respectively for isothermal and non-isothermal cases. For non-isothermal case, reaction was slower than the isothermal flow. As the reaction follows 1<sup>st</sup> order reaction, residence time is calculated from the 1<sup>st</sup> order formula. At the outlet, average cup concentration is calculated for both laminar and plug. If the reaction rate constant is increased, more scalar1 will be converted to scalar2 in the reaction. For Plug flow, velocity is flat profile velocity is same inside the contour. We found reaction-based residence time is almost same as flow based (around 5s). For both cases, we see the decay nature of scalar1 from inlet to outlet.

## References

- [1] D. K. Garg, C. A. Serra, Y. Hoarau, D. Parida, M. Bouquey, and R. Muller, “Numerical investigations of different tubular microreactor geometries for the synthesis of polymers under unmixed feed condition,” *Macromol. Theory Simul.*, vol. 29, no. 4, p. 2000008, 2020.
- [2] H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, 2nd ed. Harlow, U.K.: Pearson Education, 2007.
- [3] R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena*, 2nd ed. New York, NY, USA: Wiley, 2002.
- [4] H. Schlichting and K. Gersten, *Boundary-Layer Theory*, 9th ed. Berlin, Germany: Springer, 2017.
- [5] B. A. Finlayson, “Mixing of liquids in microfluidic devices,” Univ. Washington, 2008. [Online]. Available: [https://faculty.washington.edu/finlayso/che499/Mixing\\_Summary.pdf](https://faculty.washington.edu/finlayso/che499/Mixing_Summary.pdf)
- [6] J. Pinto and M. Sitoula, “Lab migration report on modelling passive scalar as reactive chemical species without diffusion in a laminar flow reactor following first order chemical reaction,” unpublished.
- [7] FOSSEE, *FOSSEE CFD OpenFOAM Internship Tasks and Guidelines*, 2025. [Online]. Available: <https://cfd.fossee.in/>