Pollutant Jet in Crossflow

Soravis Tangthavonsirikul 6131334521 Jirayu Hanwong 6130082121

Abstract

Computational Fluid dynamics or CFD is used for solving complex fluid problem which have many variables and high computational cost. Pollutant Jet in Crossflow is the one of the problems used CFD to solve. The objective of this project is to implement the solver in 2D and 3D and visualize variables of fluid and pollutant. To achieve the objectives, we implemented 2D solver with C++ programing language to validate, and 3D solver for jet in crossflow problem. It was found that code from 2D solver showed the results like theory. For 3D solver, we have finished the solver already, but we faced with barrier of finding the best parameters to run and computational power.

Introduction

Fluid dynamics is the branch of fluid mechanics that describes the flow of fluids - liquids and gasses. Physicists and engineers study fluid dynamics because it provides a method for studying many fields such as ocean currents, weather patterns, and blood circulation, also the application of fluid dynamics includes rocket engines, wind turbines, and air conditioning systems. To solve the problems, fluid dynamics is the calculation of different fluid properties as a function of time and space such as flow velocity, pressure, density, and temperature.

Most problems in fluid dynamics are too complex to be solved by direct calculation. In these cases, problems must be solved by numeric methods using computer simulations. This area of study is called numerical or computational fluid dynamics (CFD). CFD covers a broad range of research and technology challenges, including aerodynamics and aerospace analytics, climate solution, science, and environmental engineering.

The air pollution has become a serious environmental and health issue around the world. To study and solve the problem, CFD is used for this. A CFD solver developed by C++ programing language can conduct a 2D and 3D simulation that help depict the distribution of polluted air, being exhausted from a smoke stack.

Governing Equations

With the assumption of incompressible flow, Navier -Stokes Equations and poisson equation play an important role to govern fluid properties.

The Navier-Stokes equation consists of a time-independent continuity equation for conservation of mass, three time-dependent conservation of momentum equations in equation 1 and time-dependent conservation of energy equation in equation 2.

Continuity Equation

$$\frac{\partial}{\partial x_i^*} u_i^* = 0 \tag{1}$$

Momentum Equation

$$\frac{\partial}{\partial t^*} u_i^* + u_j^* \frac{\partial}{\partial x_j^*} u_i^* = -\frac{\partial}{\partial x_i^*} p^* + Re^{-1} \frac{\partial^2}{\partial x_j^* \partial x_j^*} u_i^* \tag{2}$$

The * superscript denotes dimensional quantities. Re is the Reynolds number. These equations are dimensionless. Neglect the external influences.

The Poisson pressure equations is derived from starting with the equation (2), then the divergence is applied to the equation and assume the fluid as incompressible fluid. The equation is shown as equation (3)

Poisson pressure equation :

$$\frac{\partial}{\partial x_j^*} u_i^* \frac{\partial}{\partial x_i^*} u_j^* = -\frac{\partial^2}{\partial x_i^* \partial x_i^*} p^* \tag{3}$$

The concentration of pollutant $(\emptyset)$ is represented as a passive scalar governed by convection-diffusion equation as equation (4).

Convection Diffusion equation :

$$\frac{\partial}{\partial t}\phi + u_j \frac{\partial}{\partial x_j}\phi = Re^{-1} \frac{\partial^2}{\partial x_j \partial x_j}\phi \tag{4}$$

Numerical Method

The numerical treatment of the Navier-Stokes equations are derived from Numerical Simulation in Fluid Dynamice by Griebel et al. [1]

When solving the Navier-Stokes equation, the region is often discretized using a staggered grid, in which the different unknown variables are not located at the same grid points as shown in Fig. 1.

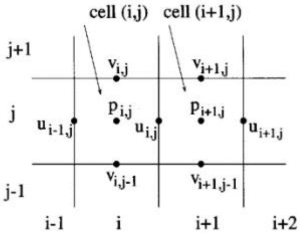

Figure 1 Staggered grid

The continuity equation is discretized at the center at the center of each cell $(i,j), i = 1, \dots, i_{max}, j = 1, \dots, j_{max}$ by replacing the spatial derivatives $\frac{\partial u}{\partial x}, \frac{\partial v}{\partial y}$ by centered differences using half the mesh width.

$$\left[\frac{\partial u}{\partial x}\right]_{i,j} := \frac{u_{i,j} - u_{i-1,j}}{\delta x}, \qquad \left[\frac{\partial v}{\partial y}\right]_{i,j} := \frac{v_{i,j} - v_{i,j-1}}{\delta y}.$$

The second derivative terms, the convective terms and the pressure to discretize them are written as shown. For u at the midpoint of right edge of cell $(i,j), i = 1, \dots, i_{max} - 1, j = 1, \dots, j_{max}$, the equations are set

$$\begin{aligned}
\left[\frac{\partial(u^2)}{\partial x}\right]_{i,j} &:= \frac{1}{\delta x}\left(\left(\frac{u_{i,j}+u_{i+1,j}}{2}\right)^2 - \left(\frac{u_{i-1,j}+u_{i,j}}{2}\right)^2\right) \\
&+ \gamma\frac{1}{\delta x}\left(\frac{|u_{i,j}+u_{i+1,j}|}{2}\frac{(u_{i,j}-u_{i+1,j})}{2} - \frac{|u_{i-1,j}+u_{i,j}|}{2}\frac{(u_{i-1,j}-u_{i,j})}{2}\right), \\
\left[\frac{\partial(uv)}{\partial y}\right]_{i,j} &:= \frac{1}{\delta y}\left(\frac{(v_{i,j}+v_{i+1,j})}{2}\frac{(u_{i,j}+u_{i,j+1})}{2} - \frac{(v_{i,j-1}+v_{i+1,j-1})}{2}\frac{(u_{i,j-1}+u_{i,j})}{2}\right) \\
&+ \gamma\frac{1}{\delta y}\left(\frac{|v_{i,j}+v_{i+1,j}|}{2}\frac{(u_{i,j}-u_{i,j+1})}{2} - \frac{|v_{i,j-1}+v_{i+1,j-1}|}{2}\frac{(u_{i,j-1}-u_{i,j})}{2}\right), \\
\left[\frac{\partial^2 u}{\partial x^2}\right]_{i,j} &:= \frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\delta x)^2}, \\
\left[\frac{\partial^2 u}{\partial y^2}\right]_{i,j} &:= \frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{(\delta y)^2}, \qquad \left[\frac{\partial p}{\partial x}\right]_{i,j} := \frac{p_{i+1,j}-p_{i,j}}{\delta x},
\end{aligned}$$

For v at the midpoint of upper edge of cell $(i,j), i = 1, \dots, i_{max}, j = 1, \dots, j_{max} - 1$, the equations are set

$$\begin{aligned}
\left[\frac{\partial(uv)}{\partial x}\right]_{i,j} &:= \frac{1}{\delta x}\left(\frac{(u_{i,j}+u_{i,j+1})}{2}\frac{(v_{i,j}+v_{i+1,j})}{2} - \frac{(u_{i-1,j}+u_{i-1,j+1})}{2}\frac{(v_{i-1,j}+v_{i,j})}{2}\right) \\
&+ \gamma\frac{1}{\delta x}\left(\frac{|u_{i,j}+u_{i,j+1}|}{2}\frac{(v_{i,j}-v_{i+1,j})}{2} - \frac{|u_{i-1,j}+u_{i-1,j+1}|}{2}\frac{(v_{i-1,j}-v_{i,j})}{2}\right), \\
\left[\frac{\partial(v^2)}{\partial y}\right]_{i,j} &:= \frac{1}{\delta y}\left(\left(\frac{v_{i,j}+v_{i,j+1}}{2}\right)^2 - \left(\frac{v_{i,j-1}+v_{i,j}}{2}\right)^2\right) \\
&+ \gamma\frac{1}{\delta y}\left(\frac{|v_{i,j}+v_{i,j+1}|}{2}\frac{(v_{i,j}-v_{i,j+1})}{2} - \frac{|v_{i,j-1}+v_{i,j}|}{2}\frac{(v_{i,j-1}-v_{i,j})}{2}\right), \\
\left[\frac{\partial^2 v}{\partial x^2}\right]_{i,j} &:= \frac{v_{i+1,j}-2v_{i,j}+v_{i-1,j}}{(\delta x)^2}, \\
\left[\frac{\partial^2 v}{\partial y^2}\right]_{i,j} &:= \frac{v_{i,j+1}-2v_{i,j}+v_{i,j-1}}{(\delta y)^2}, \qquad \left[\frac{\partial p}{\partial y}\right]_{i,j} := \frac{p_{i,j+1}-p_{i,j}}{\delta y}.
\end{aligned}$$

On the boundary, the velocities value are obtained from a discretization of the boundary conditions of continuous problem. For no-slip condition, we can set condition as

$$\begin{aligned}
u_{0,j} &= 0, & u_{i_{\max},j} &= 0, & j &= 1,\dots,j_{\max}, \\
v_{i,0} &= 0, & v_{i,j_{\max}} &= 0, & i &= 1,\dots,i_{\max}.
\end{aligned}$$

$$\begin{aligned}
v_{0,j} &= -v_{1,j}, & v_{i_{\max}+1,j} &= -v_{i_{\max},j}, & j &= 1,\dots,j_{\max}, \\
u_{i,0} &= -u_{i,1}, & u_{i,j_{\max}+1} &= -u_{i,j_{\max}}, & i &= 1,\dots,i_{\max}.
\end{aligned}$$

For free-slip condition, we can set condition as

$$\begin{aligned}
u_{0,j} &= 0, & u_{i_{\max},j} &= 0, & j &= 1,\dots,j_{\max}, \\
v_{i,0} &= 0, & v_{i,j_{\max}} &= 0, & i &= 1,\dots,i_{\max}.
\end{aligned}$$

$$\begin{aligned}
v_{0,j} &= v_{1,j}, & v_{i_{\max}+1,j} &= v_{i_{\max},j}, & j &= 1,\dots,j_{\max}, \\
u_{i,0} &= u_{i,1}, & u_{i,j_{\max}+1} &= u_{i,j_{\max}}, & i &= 1,\dots,i_{\max}.
\end{aligned}$$

For outflow condition, we can set condition as

$$\left.\begin{aligned}
u_{0,j} &= u_{1,j}, & u_{i_{\max},j} &= u_{i_{\max}-1,j}, \\
v_{0,j} &= v_{1,j}, & v_{i_{\max}+1,j} &= v_{i_{\max},j},
\end{aligned}\right. \quad j = 1,\dots,j_{\max},$$

$$\left.\begin{aligned}
u_{i,0} &= u_{i,1}, & u_{i,j_{\max}+1} &= u_{i,j_{\max}}, \\
v_{i,0} &= v_{i,1}, & v_{i,j_{\max}} &= v_{i,j_{\max}-1},
\end{aligned}\right. \quad i = 1,\dots,i_{\max}.$$

For inflow condition, we impose this for the velocities normal to the boundary by directly fixing the boundary on the boundary line.

To discretize the time derivatives at time t_{n+1} , we use Euler's method as shown below.

$$\left[\frac{\partial u}{\partial t}\right]^{(n+1)} := \frac{u^{(n+1)} - u^{(n)}}{\delta t}, \qquad \left[\frac{\partial v}{\partial t}\right]^{(n+1)} := \frac{v^{(n+1)} - v^{(n)}}{\delta t},$$

Where the superscript (n) denotes the time level.

For concentration of pollutant, it can be described in space and time discretization as well.

$$\left[\frac{\partial \phi}{\partial t}\right]_{i,j}^{(n+1)} = \frac{1}{\delta t}\left(\phi_{i,j}^{(n+1)} - \phi_{i,j}^{(n)}\right)$$

$$\begin{aligned}\left[\frac{\partial u\phi}{\partial x}\right]_{i,j} = \frac{1}{\delta x}\left(u_{i,j}\frac{\phi_{i,j} + \phi_{i+1,j}}{2} - u_{i-1,j}\frac{\phi_{i-1,j} + \phi_{i,j}}{2}\right) \\ + \frac{\gamma}{\delta x}\left(\left|u_{i,j}\right|\frac{\phi_{i,j} - \phi_{i+1,j}}{2}\right. \\ \left. - \left|u_{i-1,j}\right|\frac{\phi_{i-1,j} + \phi_{i,j}}{2}\right)\end{aligned}$$

$$\begin{aligned}\left[\frac{\partial v\phi}{\partial y}\right]_{i,j} = \frac{1}{\delta y}\left(v_{i,j}\frac{\phi_{i,j} + \phi_{i,j+1}}{2} - v_{i-1,j}\frac{\phi_{i,j-1} + \phi_{i,j}}{2}\right) \\ + \frac{\gamma}{\delta y}\left(\left|v_{i,j}\right|\frac{\phi_{i,j} - \phi_{i,,j+1}}{2}\right. \\ \left. - \left|v_{i,j-1}\right|\frac{\phi_{i,j-1} + \phi_{i,j}}{2}\right)\end{aligned}$$

$$\left[\frac{\partial^2 \phi}{\partial x^2}\right]_{i,j} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{\delta x^2}$$

$$\left[\frac{\partial^2 \phi}{\partial y^2}\right]_{i,j} = \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{\delta y^2}$$

Beginning at time $t = 0$ with given initial values for velocities, time is increased by small time changing in each step until the final time is reached. At every time step, all variables from previous time step are known and those are to be computed at present time step.

We can discretize time from $\frac{\partial u}{\partial t}$ and $\frac{\partial v}{\partial t}$ in the momentum equations.

$$\begin{aligned} u^{(n+1)} &= u^{(n)} + \delta t\left[\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{\partial(u^2)}{\partial x} - \frac{\partial(uv)}{\partial y} + g_x - \frac{\partial p}{\partial x}\right] \\ v^{(n+1)} &= v^{(n)} + \delta t\left[\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \frac{\partial(uv)}{\partial x} - \frac{\partial(v^2)}{\partial y} + g_y - \frac{\partial p}{\partial y}\right] \end{aligned}$$

To separate pressure terms apart from the others, we introduce F and G as

$$\begin{aligned} F &:= u^{(n)} + \delta t\left[\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{\partial(u^2)}{\partial x} - \frac{\partial(uv)}{\partial y} + g_x\right], \\ G &:= v^{(n)} + \delta t\left[\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \frac{\partial(uv)}{\partial x} - \frac{\partial(v^2)}{\partial y} + g_y\right], \end{aligned}$$

While the F and G terms are evaluated at time level n, $\frac{\partial p}{\partial x}$ and $\frac{\partial p}{\partial y}$ are associated with time level n+1. We can describe it as the time discretization of the momentum equations as shown below.

$$\begin{aligned} u^{(n+1)} &= F^{(n)} - \delta t\,\frac{\partial p^{(n+1)}}{\partial x}, \\ v^{(n+1)} &= G^{(n)} - \delta t\,\frac{\partial p^{(n+1)}}{\partial y}. \end{aligned}$$

To solve the pressure at time step n+1, we can compute it from Poisson equation.

$$\frac{\partial^2 p^{(n+1)}}{\partial x^2} + \frac{\partial^2 p^{(n+1)}}{\partial y^2} = \frac{1}{\delta t}\left(\frac{\partial F^{(n)}}{\partial x} + \frac{\partial G^{(n)}}{\partial y}\right).$$

To solve the problems, these are the steps how we implemented it. First, we assigned the initial value of the velocities and pressure. Then, we started setting the while loop. In each time step, the boundary values had been set before computing F and G value at this time step.

$$F_{i,j} := u_{i,j} + \delta t \left(\frac{1}{Re} \left(\left[\frac{\partial^2 u}{\partial x^2}\right]_{i,j} + \left[\frac{\partial^2 u}{\partial y^2}\right]_{i,j} \right) - \left[\frac{\partial (u^2)}{\partial x}\right]_{i,j} - \left[\frac{\partial (uv)}{\partial y}\right]_{i,j} + g_x \right)$$

$$i = 1, \ldots, i_{\max} - 1, \quad j = 1, \ldots, j_{\max},$$

$$G_{i,j} := v_{i,j} + \delta t \left(\frac{1}{Re} \left(\left[\frac{\partial^2 v}{\partial x^2}\right]_{i,j} + \left[\frac{\partial^2 v}{\partial y^2}\right]_{i,j} \right) - \left[\frac{\partial (uv)}{\partial x}\right]_{i,j} - \left[\frac{\partial (v^2)}{\partial y}\right]_{i,j} + g_y \right)$$

$$i = 1, \ldots, i_{\max}, \quad j = 1, \ldots, j_{\max} - 1 .$$

F and G values were computed as an right-hand side (RHS) term to solve the Poisson equation.

$$RHS_{i,j} = \frac{1}{\delta t}\left(\frac{F_{i,j}^{(n)} - F_{i-1,j}^{(n)}}{\delta x} - \frac{G_{i,j}^{(n)} - G_{i,,j-1}^{(n)}}{\delta y} \right)$$

To solve the Poisson equation, the Successive overrelaxation (SOR) iteration method is an algorithm to solve the pressure.

$$it = 1, \ldots, it_{\max}, \quad i = 1, \ldots, i_{\max}, \quad j = 1, \ldots, j_{\max},$$

$$p_{i,j}^{\mathrm{it}+1} := (1-\omega)\, p_{i,j}^{\mathrm{it}} + \frac{\omega}{\left(\frac{\epsilon_i^E + \epsilon_i^W}{(\delta x)^2} + \frac{\epsilon_j^N + \epsilon_j^S}{(\delta y)^2} \right)} \cdot \left(\frac{\epsilon_i^E p_{i+1,j}^{\mathrm{it}} + \epsilon_i^W p_{i-1,j}^{\mathrm{it}+1}}{(\delta x)^2} + \frac{\epsilon_j^N p_{i,j+1}^{\mathrm{it}} + \epsilon_j^S p_{i,j-1}^{\mathrm{it}+1}}{(\delta y)^2} - rhs_{i,j} \right). \tag{3.44}$$

$$\epsilon_i^W := \begin{cases} 0, & i = 1, \\ 1, & i > 1, \end{cases} \quad \epsilon_i^E := \begin{cases} 1, & i < i_{\max}, \\ 0, & i = i_{\max}, \end{cases} \quad \epsilon_j^S := \begin{cases} 0, & j = 1, \\ 1, & j > 1, \end{cases} \quad \epsilon_j^N := \begin{cases} 1, & j < j_{\max}, \\ 0, & j = j_{\max}, \end{cases}$$

The iteration is terminated either once a maximal number of steps maximum iteration had been taken or when the norm of residual has fallen below an absolute tolerance assigned by us.

$$r_{i,j}^{\mathrm{it}} := \frac{\epsilon_i^E (p_{i+1,j}^{\mathrm{it}} - p_{i,j}^{\mathrm{it}}) - \epsilon_i^W (p_{i,j}^{\mathrm{it}} - p_{i-1,j}^{\mathrm{it}})}{(\delta x)^2} + \frac{\epsilon_j^N (p_{i,j+1}^{\mathrm{it}} - p_{i,j}^{\mathrm{it}}) - \epsilon_j^S (p_{i,j}^{\mathrm{it}} - p_{i,j-1}^{\mathrm{it}})}{(\delta y)^2} - rhs_{i,j}, \tag{3.45}$$

$$i = 1, \ldots, i_{\max}, \quad j = 1, \ldots, j_{\max},$$

$$\|r^{\mathrm{it}}\|_2 := \left(\frac{1}{i_{\max} j_{\max}} \sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} (r_{i,j}^{\mathrm{it}})^2 \right)^{1/2}$$

After calculating pressure at the new time step using SOR algorithm, the velocities at the new time step could be computed from the equation.

$$u_{i,j}^{(n+1)} = F_{i,j}^{(n)} - \frac{\delta t}{\delta x}(p_{i+1,j}^{(n+1)} - p_{i,j}^{(n+1)}),$$

$$i = 1, \ldots, i_{\max} - 1, \quad j = 1, \ldots, j_{\max},$$

$$v_{i,j}^{(n+1)} = G_{i,j}^{(n)} - \frac{\delta t}{\delta y}(p_{i,j+1}^{(n+1)} - p_{i,j}^{(n+1)}),$$

$$i = 1, \ldots, i_{\max}, \quad j = 1, \ldots, j_{\max} - 1,$$

Finally, to make solver stability, the time step changing could be adjusted as a dynamic value based on the condition or a constant value as long as it fell under this condition.

$$\delta t := \tau \min \left(\frac{Re}{2} \left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2} \right)^{-1}, \frac{\delta x}{|u_{\max}|}, \frac{\delta y}{|v_{\max}|} \right).$$

These variables were repeatedly calculated until it reached final time step.

Validation

Before starting 3D simulation of jet in crossflow (JICF), we validated the 2D solver using setting initial condition of concentration of pollutant = 1 for the lower half of inlet flow and 0 elsewhere. For air, we set initial velocities in vertical axis, Reynold number = 300. Where $dx = 0.1, dy = 0.3$, $\delta t = 0.01, N = 1{,}111$, we run the simulation until value of velocities and pressure less changed.

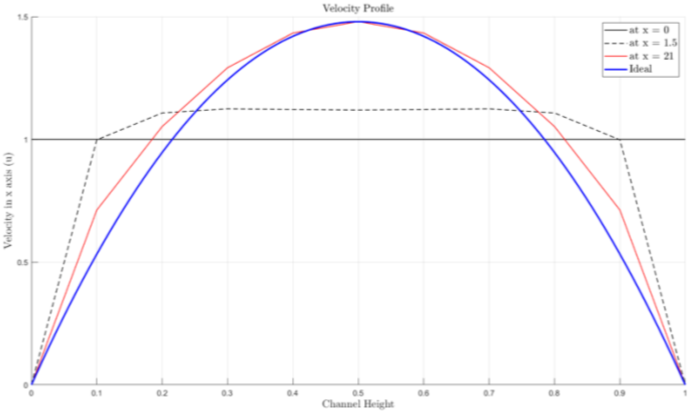

Figure 2 The velocities profile across channel

Fig 2 illustrates the temporal-mean streamwise velocity profile across the channel in the x-axis. The different lines show the velocity profile at different position. The black, dot and red lines were the results that were simulated by the solver. The blue line was the result from theory. We can see that the result from simulation was close to the theory when it was far enough from inlet pipe.

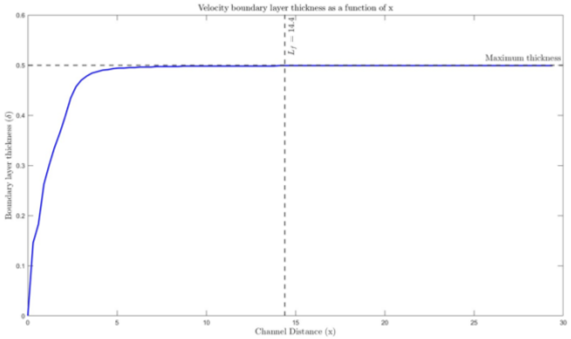

Figure 4 the velocity boundary layer thickness over the channel

Fig 3 illustrates the velocity boundary layer thickness over the channel distance. When distance was far from the inlet, the boundary layer thickness grew over the distance, like theory. From the graph, a fully develop distance is 14.4 from inlet which is nearly entrance length in theory = 15.

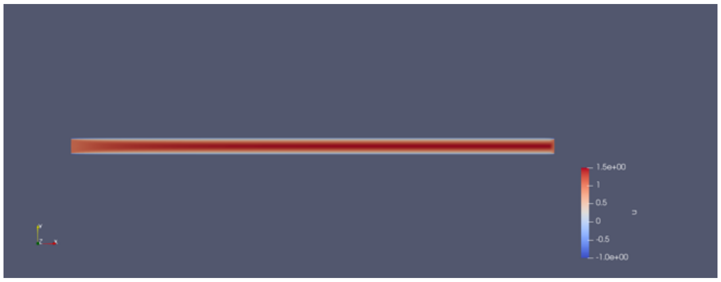

Figure 5 vertical velocity contour

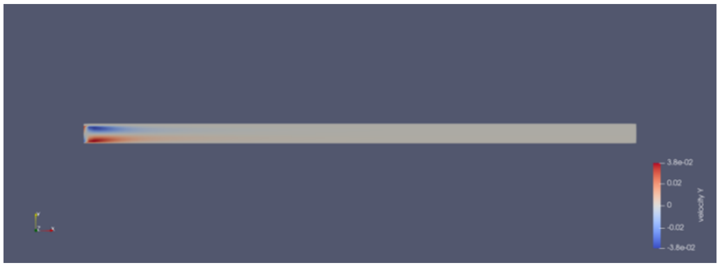

Figure 6 horizontal velocity contour

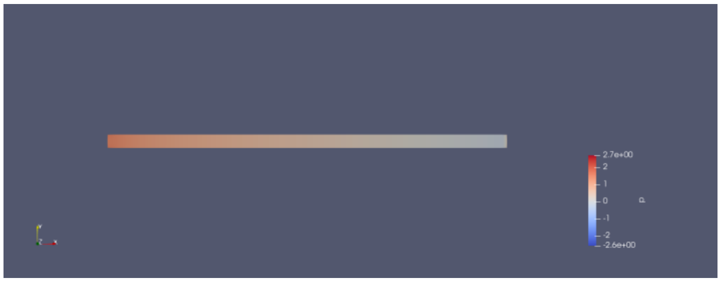

Figure 3 Pressure contour

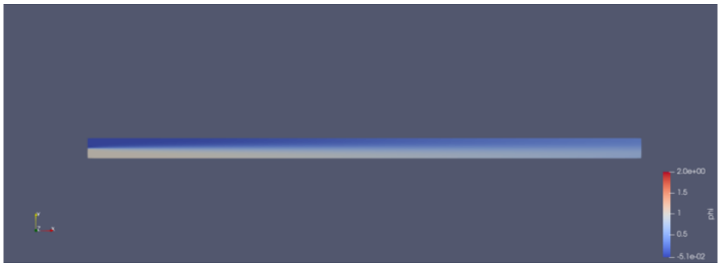

Figure 7 Pollutant concentration contour

Figure 4, 5, 6 ,and 7 shows the velocity, pressure and concentration of pollutant in the final time step, respectively. In fig 6, there was the gradient of pressure between inlet of pipe and inside of it because the fluid flowed. In fig 7, the concentration of pollutant diffused from left to right because of the velocity of air in vertical direction and it was diluted when far from the inlet of pipe. In addition, the little pollutant was diffused in vertical axis.

Jet in cross flow (JICF)

For 3D simulation, we implemented the code by extending it from 2D code. Use the computational domain size as what is shown in the figure 8, Dirichlet inflow boundary condition for the inlet, No slip boundary condition at the bottom boundary and Neumann outflow condition elsewhere. Set the Reynolds number = 1000 and the ratio of horizontal jet velocity and crossflow velocity is 2. With the number of grid points over million points, it took high computational cost and time. So, we tried our best to find the parameters that could run the simulation as long as possible (200 time steps) before the result blew up. The result of streamline has shown in figure 8. The result shows that the air flowed from left to right and had vertical jet from the under.

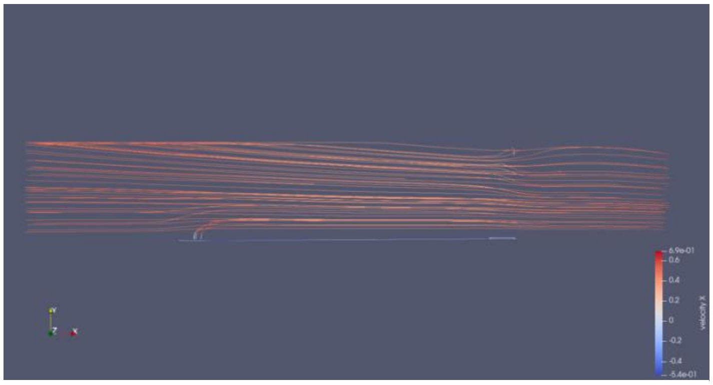

Figure 8 3D Streamline

The pressure and concentration of pollutant of 3D simulation are shown in figure 9 and 10. We didn't see much change from the simulation because the value blew off. The elliptic hole under the domain came from adjusting length of grid in x axis greater than z axis to prevent the blew off of simulation.

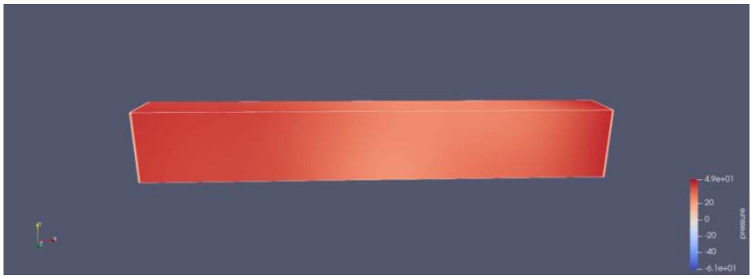

Figure 9 3D pressure contour

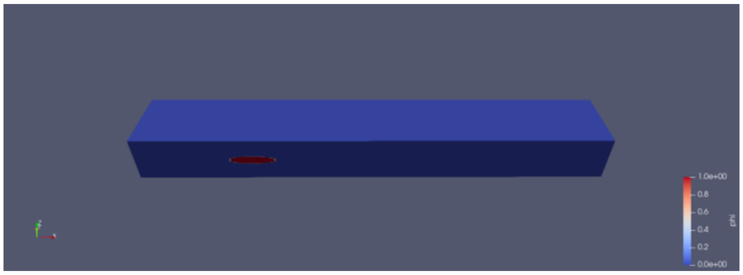

Figure 10 Pollutant's concentration contour

Even if we could not simulate 3D solver to the time = 50, we show the method how to analyze below if we had simulation results.

- Measured the diameter of the pollutant cloud to find safe zone that didn't affect from pollutant
- Because of vertical velocity, the pollutant could be diffused and we could identify area and distance that household area should be placed to prevent people from pollutant.

Conclusion

The 2D solver showed the result as the theory shows and the 3D solver could be do better if we find better parameter such as grid size, time step size and the maximum number of iteration for our computer.

Reference

[1] T. D. T. N. Micheal Griebel, Numerical Simulation in Fluid Dynamics : a practical introduction, Siam, 1997.