Numerical Analysis of Droplet Distribution in Spray Painting

Shyam Sundar Hemamalini (5071984)
Ravi Ramesh (5020336)

Delft University of Technology, The Netherlands

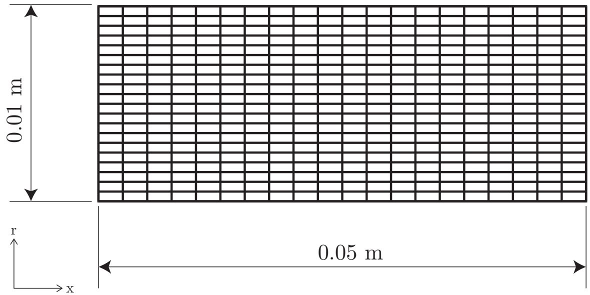
July 30, 2020

Introduction

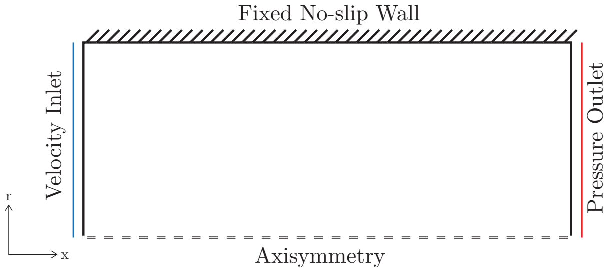
- One-way coupled Euler-Lagrange multiphase solver was coded in Python.
- 2D Prandtl mixing length based RANS solver was built for the continuous phase simulation with Nikuradse's formula for mixing length.
- Discrete Eddy Simulation combined with Monte Carlo Simulation solver was built for the solution of dispersed phase.
- Fluid side data was stored as CSV files in axisymmetric form which was then converted and interpolated.
- The code is tested for a paint spray with $\rho_p = 1200\ \mathrm{kg/m^3}$ and $\rho = 1.25\ \mathrm{kg/m^3}$.
- Scope for improvement of code mentioned.

Continuous Phase - Geometry & Mesh

- Narrow cylindrical pipe of length 50 mm, diameter 20 mm.
- Flow assumed to have circumferential and axisymmetric symmetry and hence, geometry is modelled in 2D from $r = 0$ to $r = 10$ mm.
- Uniform mesh of 20 grid-steps in both axes.



Continuous Phase - Boundary Conditions



- As the solver employs a staggered grid for U and p and works on the finite volume method, implementation of boundary conditions require the use of ghost cells.

Boundary Conditions - Velocity

Boundary	Boundary Condition	
	Type	Value
Inlet	Dirichlet	$u = 5$ m/s
	Dirichlet	$v = 0$
Outlet	Neumann	$\mathrm{grad}(\vec{U}) = 0$
No slip wall	Dirichlet	$\vec{U} = 0$
Axisymmetry	Neumann	$\mathrm{grad}(u) = 0$
	Dirichlet	$v = 0$

- At the inlet, ghost cells are required for v.
- At the outlet, ghost cells are required for both u and v.
- At the fixed wall, ghost cells are required for u.
- At the axisymmetry boundary, ghost cells are required for u.

Boundary Conditions - Pressure

Boundary	Boundary Condition	
	Type	Value
Inlet	Neumann	$\mathrm{grad}(p) = 0$
Outlet	Dirichlet	$p = 10^5$ Pa
No slip wall	-	-
Axisymmetry	Neumann	$\mathrm{grad}(p) = 0$

- Ghost cells are required at all the four boundaries.
- At the fixed wall, the method described by Tryggvason is followed, where the pressure at the ghost cells is given a value $p = 0$ and the density at the ghost cell is given a very high value, so as to numerically remove the term from the discretised equations.

Continuous Phase - Solution Algorithm

- The Navier-Stokes equations are solved for a finite time without the turbulence model. After a distinct profile is developed, the turbulence model is then turned on.

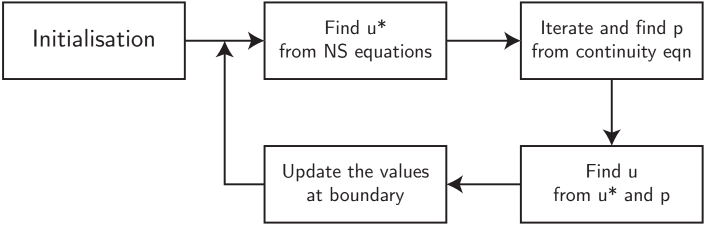


Figure: Navier-Stokes Predictor-Corrector Algorithm

Continuous Phase - Turbulence Model

- A 2D Prandtl mixing length RANS model with Smagorinsky's relation for eddy viscosity as below was used.

$$\nu_t = L_m^2 \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2}$$

- The mixing length is given by Nikuradse's formula for pipe flows as

$$L_m = r_0 \left(0.14 - 0.08\left(1 - \frac{y}{r_0}\right)^2 - 0.06\left(1 - \frac{y}{r_0}\right)^4\right)$$

- Viscosity values stored in the same manner as pressure at the cell centers with the Neumann boundary condition $\mathrm{grad}(\nu) = 0$ imposed at all the boundaries.

Continuous Phase - Solution Algorithm

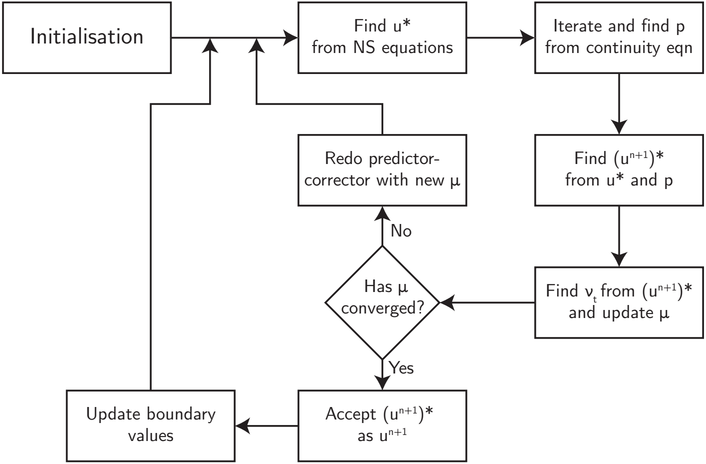
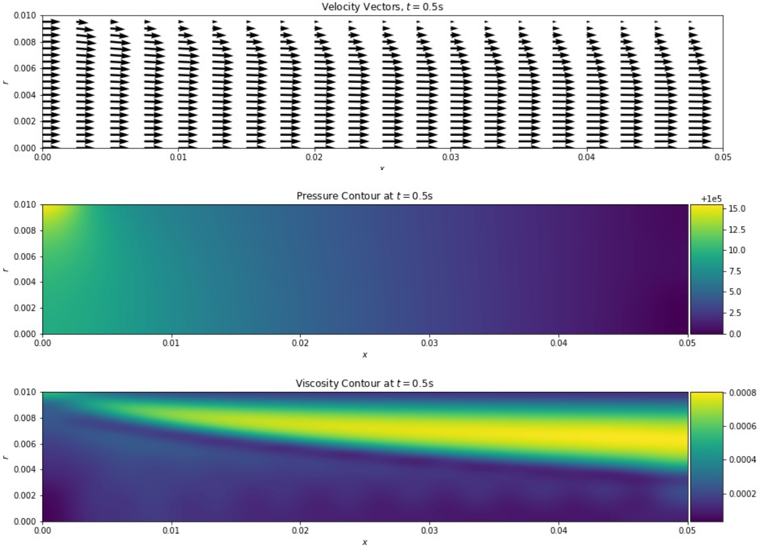


Figure: Turbulence Model Algorithm

Continuous Phase - Execution & Exporting

- Adaptive time-stepping algorithm was used.
    - The solution was tested for a particular dt.
    - If the solution diverges, dt was reduced and the solver was ran again.
    - If the solution converged too rapidly, dt was increased for the next time-step.
- The velocity, pressure and viscosity data from each time-step were stored as CSV files along with flow time and dt data.
    - This is required for the dispersed phase coupling.
    - If the solver crashed, this data was used to resume the solution from the last stable point.
- The solver was programmed and executed in Google Colaboratory which allowed data to be routinely saved in Google Drive, which also served as backup.

Continuous Phase - Example Output



Dispersed Phase - Relevant Assumptions

- Neglect the effect of gravity as the flow is highly inertial.
- Neglect the effect of Basset history force.
- Neglect the effect of Saffman Lift as $Re_p \gg 1$.
- Neglect the effect of particle collisions since volume fraction is low.
- Neglect the effect of added mass and buoyancy force since $\rho \ll \rho_p$.
- Specular, perfectly elastic reflections from the wall.
- Drag force acting on particle due to viscous forces, given by drag law.
- Isotropic turbulence assumed while calculating u'.

Dispersed Phase - Handling Continuous Phase Data

- Convert axisymmetric data into cartesian form.
- Reference the data using proper names from Google Drive.
- Use bilinear interpolation to compute the fluid properties at the position of the particle.

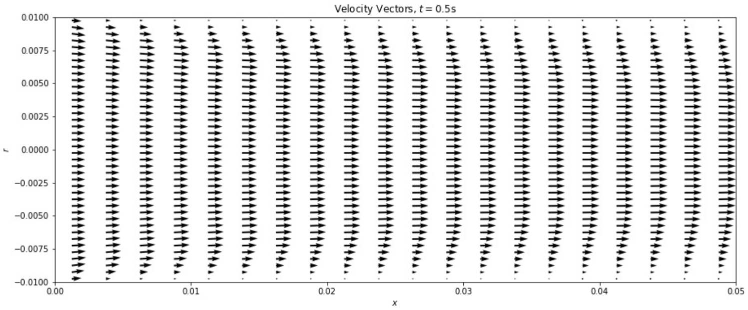


Figure: Axisymmetric Data in Cartesian Form

Dispersed Phase - Discrete Eddy Simulation

Since the flow is turbulent, addition of velocity fluctuations is required. This is done through the Discrete Eddy Simulation method.

- The scale of velocity fluctuations is given by

$$U' = L_m \left| \frac{\partial U}{\partial y} \right|$$

- Since we already have $\nu_t = L_m^2 \left| \frac{\partial U}{\partial y} \right|$ with the same method, we can then state

$$U' = \frac{\nu_t}{L_m}$$

- This is taken as the standard deviation σ for the fluctuations with mean $\overline{U}$ and a Gaussian distribution is considered to determine U'. Hence,

$$U = \overline{U} + \mathcal{N}(\mu = \overline{U}, \sigma = U')$$

Dispersed Phase - Simulation Algorithm

- The particle Reynolds number is computed at each time step as:

$$Re_p^n = \frac{\rho \left|\vec{V}^n - \vec{U}^n\right| D_p}{\mu}$$

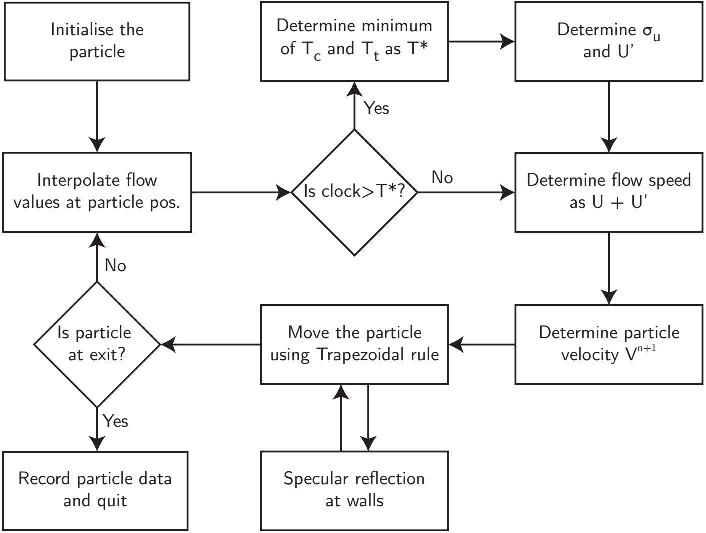
- The particle relaxation time at each time step is computed by using the following equation:

$$\tau_p^n = \frac{\rho_p}{18\,\mu} \frac{D_p^2}{(1 + 0.15\,(Re_p^n)^{0.687}))}$$

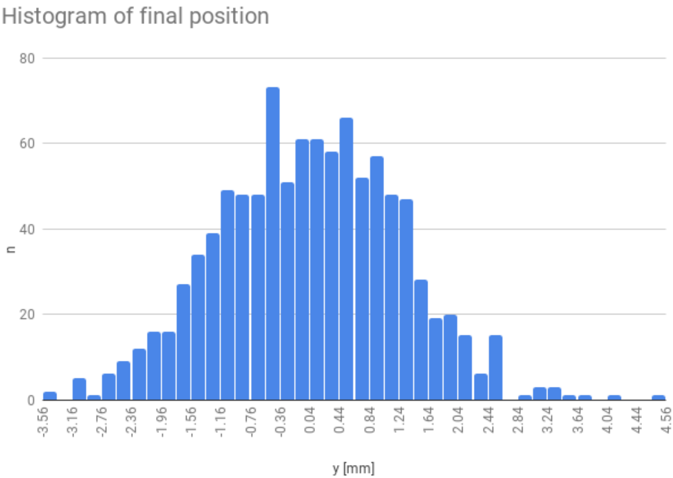
- The particle velocity is then computed by using a Forward-Euler time stepping scheme as:

$$\vec{V}^{n+1} = \vec{V}^n - \Delta t \frac{\vec{V}^n - \vec{U}^n}{\tau_p}$$

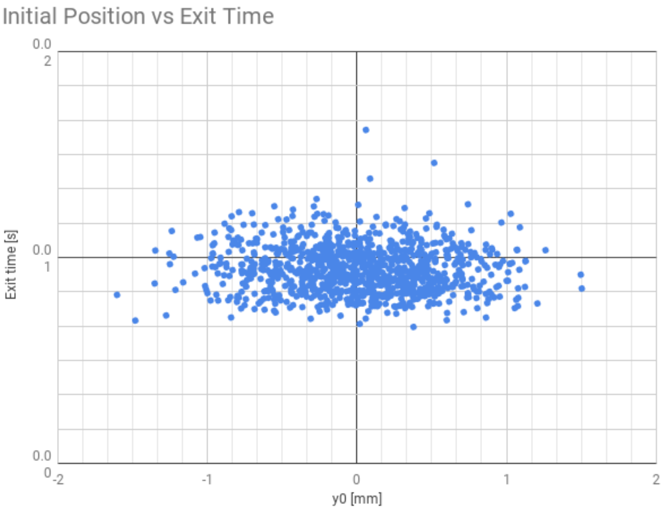
Dispersed Phase - Simulation Algorithm



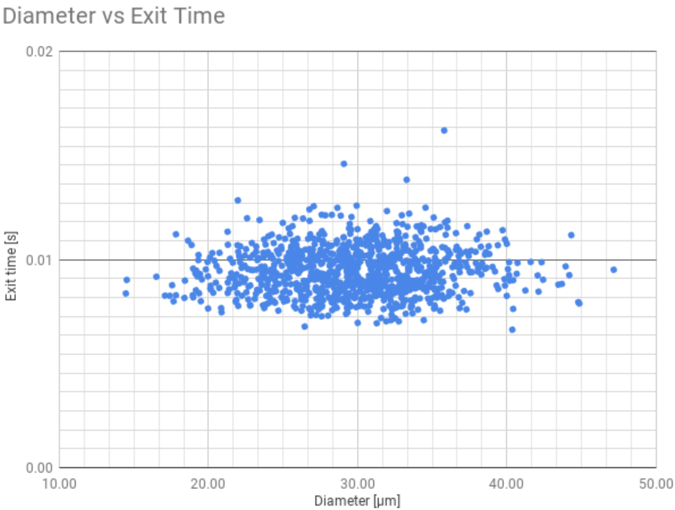
Results & Discussion



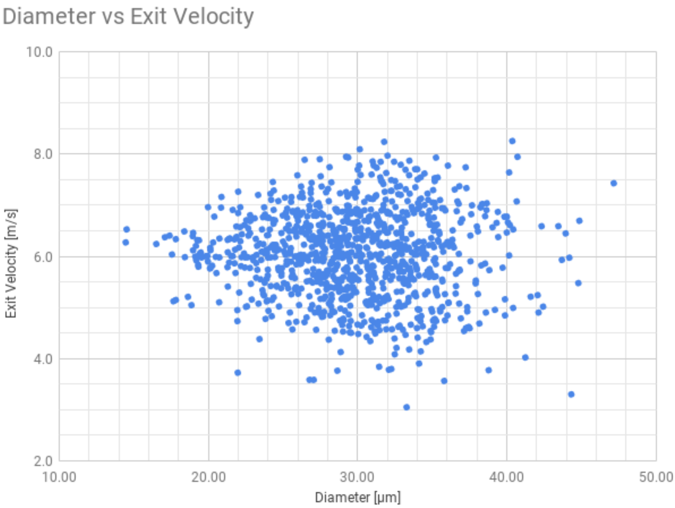
Results & Discussion



Results & Discussion



Results & Discussion



Conclusion

- Code covers dispersion of particles from a given location in a spray pipe, including the effects of turbulent and viscous dispersion.
- Fluid side solver that runs on Prandtl mixing length model along with the Nikuradse's relation between spatial location and the mixing length.
- Dispersed phase solver that includes various particle motion parameters such as specular reflection, particle deletion at the end of the pipe, bilinear interpolation of fluid velocities are added in separate modules.
- Modularity of code that allows for addition of further modules like collisions and coalescence.
- Distribution of particle position and velocity at the pipe outlet discussed.

Scope for Improvement

- Inclusion of specular reflection with inelastic collisions with the wall.
- Inclusion of particle collisions (elastic/inelastic) and coalescence modules.
- Better representation of the boundary layer (including modelling of buffer layer and log-law layer) and finer mesh resolution.
- Effects of anisotropy can be incorporated by introducing models to relate different fluctuating velocity components.