

Interactive Pipe Flow Simulation: Report & Documentation

1. Introduction

This report documents an interactive pipe flow simulation developed for visualizing and analyzing fluid behavior in pipes with varying cross-sectional areas. The simulation demonstrates fundamental fluid dynamics principles, particularly focusing on the continuity equation and Bernoulli's principle.

2. Theoretical Background

2.1. Continuity Equation

In fluid dynamics, the continuity equation expresses the conservation of mass in a flowing fluid. For an incompressible fluid in a pipe with no branches, the product of cross-sectional area and velocity must remain constant along the pipe:

$$A_1 v_1 = A_2 v_2$$

Where:

- A_1 , A_2 are cross-sectional areas at points 1 and 2
- v_1 , v_2 are fluid velocities at points 1 and 2

This means that when a pipe narrows, the fluid velocity must increase to maintain the same mass flow rate, and conversely, when a pipe widens, the fluid velocity decreases.

2.2. Bernoulli's Principle

Bernoulli's principle relates pressure, velocity, and elevation in a moving fluid. For a horizontal pipe with constant elevation, the principle can be simplified to:

$$P_1 + \frac{1}{2}\rho v_1^2 = P_2 + \frac{1}{2}\rho v_2^2$$

Where:

- P_1 , P_2 are pressures at points 1 and 2
- ρ is the fluid density
- v_1 , v_2 are fluid velocities at points 1 and 2

This equation shows that in regions of high velocity, pressure decreases, and vice versa. This principle

explains why pressure drops in narrowed sections of pipe where velocity increases.

3. Implementation Details

3.1. Class Structure

The simulation is implemented in the `InteractivePipeFlowSimulation` class with the following key components:

1. **Initialization**
 - Sets up initial parameters (velocity, density)
 - Prepares data structures for pipe sections
2. **Pipe Section Management**
 - Methods to add and configure pipe sections
 - Automatic positioning and connection of sections
3. **Physics Calculations**
 - Implementation of continuity equation for velocity calculations
 - Implementation of Bernoulli's equation for pressure calculations
4. **Particle System**
 - Particle initialization with random positions
 - Position updates based on local velocity
 - Boundary handling and recycling of particles
5. **Visualization**
 - Interactive matplotlib-based interface
 - Animated particles using `FuncAnimation`
 - Dynamic graphs with real-time updates
 - Interactive sliders for parameter adjustments

3.2. Key Methods

- `add_pipe_section()`: Adds a new pipe segment with specified dimensions
- `calculate_velocities()`: Computes fluid velocity in each section based on continuity
- `calculate_pressures()`: Determines pressure in each section using Bernoulli's equation
- `initialize_particles()`: Creates and positions particles for flow visualization

- `update_particles()`: Moves particles based on local flow velocity
- `create_interactive_visualization()`: Sets up the interactive UI with animations and sliders

4. Visualization Features

4.1. Graphical Elements

1. **Pipe Representation**

- Rectangular sections with appropriate dimensions
- Visual indication of diameter changes

2. **Particle Animation**

- Color-coded particles representing the fluid
- Motion speed proportional to local fluid velocity
- Particle recycling when exiting the pipe

3. **Dynamic Graphs**

- Real-time velocity profile along the pipe length
- Corresponding pressure distribution
- Automatic scaling of axes based on current values

4.2. Interactive Controls

1. **Velocity Slider**

- Adjusts initial flow velocity (0.5 - 5.0 m/s)
- Immediately updates all calculations and visualizations

2. **Density Slider**

- Controls fluid density (800 - 1200 kg/m³)
- Affects pressure calculations based on Bernoulli's equation

5. Example Simulation Setup

The default configuration demonstrates several key fluid dynamics concepts:

1. **Pipe Configuration**

- Initial section: length 2m, diameter 0.3m
- Constriction: length 1m, diameter 0.15m

- Recovery section: length 2m, diameter 0.3m
- Expansion: length 1m, diameter 0.45m
- Final section: length 2m, diameter 0.3m

2. **Observable Phenomena**

- Increased velocity in constricted sections
- Decreased pressure in high-velocity regions
- Recovery of pressure in expanded sections
- Conservation of mass throughout the system

6. Mathematical Implementation

6.1. Velocity Calculation

```
```python
def calculate_velocities(self):
 if not self.sections:
 return

 initial_area = self.sections[0]['area']
 initial_velocity = self.initial_velocity

 for section in self.sections:
 section['velocity'] = initial_velocity * (initial_area / section['area'])
    ```
```

This implementation directly applies the continuity equation, calculating each section's velocity based on the initial flow rate and the ratio of areas.

6.2. Pressure Calculation

```
```python
def calculate_pressures(self):
 if not self.sections:
 return

 self.calculate_velocities()

 initial_pressure = self.p0
 initial_velocity = self.sections[0]['velocity']
```

for section in self.sections:

dynamic\_pressure = 0.5 \* self.density \* (initial\_velocity\*\*2 - section['velocity']\*\*2)

section['pressure'] = initial\_pressure + dynamic\_pressure

...

This method implements Bernoulli's equation to determine pressure changes based on velocity differences, accounting for the dynamic pressure term.

## 7. User Guide

### 7.1. Running the Simulation

1. Initialize the simulation with desired parameters
2. Add pipe sections with appropriate dimensions
3. Call `create_interactive_visualization()` to launch the interface
4. Use sliders to experiment with different flow conditions

### 7.2. Interpreting Results

- **Particle Speed**: Indicates local fluid velocity
- **Color Intensity**: Represents particle properties
- **Velocity Graph**: Shows speed profile along pipe length
- **Pressure Graph**: Illustrates pressure variations
- **Slider Effects**: Observe how changing parameters affects system behavior

### 7.3. Educational Applications

- Demonstrate the relationship between pipe diameter and fluid velocity
- Visualize the inverse relationship between velocity and pressure
- Explore the conservation of mass in fluid systems
- Examine how fluid properties affect flow characteristics

## 8. Limitations and Future Work

### 8.1. Current Limitations

- Assumes ideal, incompressible fluid
- Neglects viscous effects and friction losses
- Simplified 2D representation of 3D flow
- Does not account for turbulence or boundary layer effects

## 8.2. Potential Enhancements

- Add viscosity and friction loss calculations
- Implement more complex pipe geometries (bends, branches)
- Include temperature effects and compressible flow models
- Add more detailed analytics and quantitative measurements
- Extend to 3D visualization capabilities

## 9. Conclusion

This interactive pipe flow simulation provides an accessible way to visualize and explore fundamental fluid dynamics principles. By combining theoretical calculations with an intuitive interface, it serves as an effective educational tool for understanding how fluid behaves when flowing through pipes of varying diameters.

The implementation demonstrates the practical application of the continuity equation and Bernoulli's principle, showing how these fundamental laws manifest in observable flow behavior. Through interactive manipulation of key parameters, users can develop an intuitive understanding of fluid dynamics concepts that might otherwise remain abstract.