

# Измерение расхода затопленной струи

Кирсанов Б., Ромашков С., Ражина В.

МФТИ

2025

- 1 Введение
- 2 Теория
- 3 Методика измерений
- 4 Обработка данных
- 5 Результаты
- 6 Выводы

## Цель

Определить зависимость расхода затопленной струи от расстояния до сопла

## Задачи

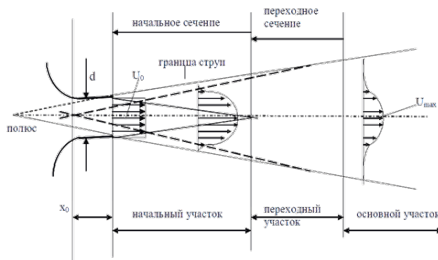
- Изучить функционал программ и написать программу для считывания данных
- Провести калибровку датчиков
- Измерить зависимость давления от расстояния
- Построить графики профилей скорости и расхода

## Затопленная струя

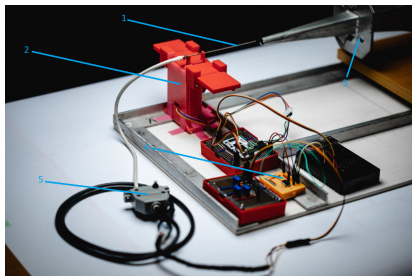
Течение, возникающее при истечении в покоящуюся окружающую среду потока жидкости или газа

## Свободное течение

Течение, при котором твердые стенки находятся на большом расстоянии и не оказывают влияния



# Экспериментальная установка



- 1 - Трубка Пито
- 2 - Шаговый двигатель
- 3 - Центробежный вентилятор
- 4 - АЦП
- 5 - Датчик давления

## Калибровка шагового двигателя

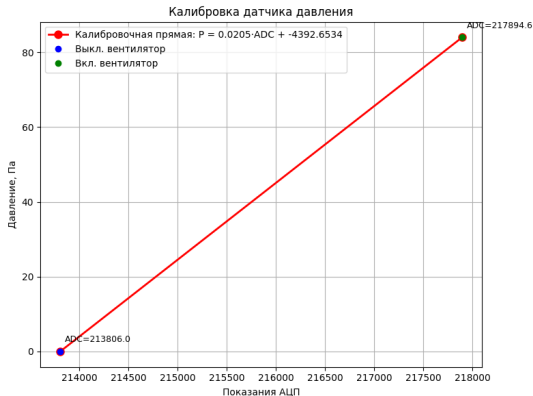
Коэффициент: 0.005 см/шаг (20 шагов/мм)



# Калибровка датчика давления

## Калибровочная зависимость

$$P = 0.0205 \cdot ADC - 4392.6534 \text{ Па}$$



# Программа эксперимента

- 1 Измерение фонового давления (вентилятор выключен)
- 2 Калибровка по цифровому манометру
- 3 Измерения в сечениях: 0, 10, 20, 30, 40, 50, 60, 70 мм
- 4 По 100 точек на каждое сечение



# Центрирование профилей скорости

## Алгоритм центрирования

- 1 Найти позицию максимальной скорости: `np.argmax(velocities)`
- 2 Вычислить смещение: `center_position = positions[max_vel_index]`
- 3 Сдвинуть позиции: `centered_positions = positions - center_position`

```
def center_velocity_profile(positions, velocities):  
  
    max_vel_index = np.argmax(velocities)  
    center_position = positions[max_vel_index]  
    centered_positions = positions - center_position  
  
    return centered_positions, velocities
```

# Вычисление расхода

## Формула расхода

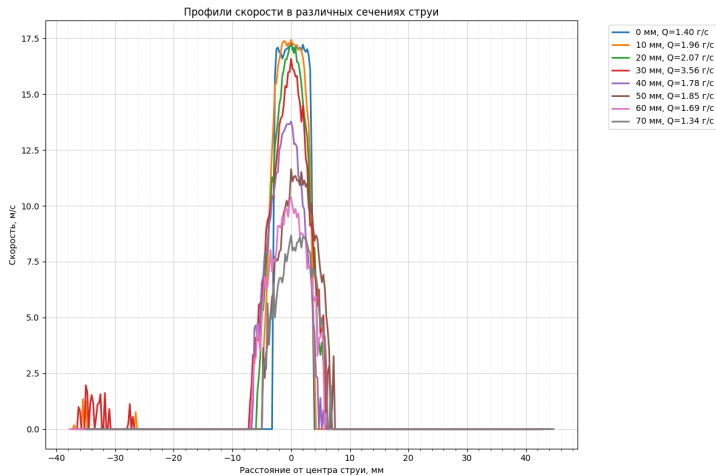
$$Q = 2\pi \int V(r) \cdot r dr$$

## Численное интегрирование

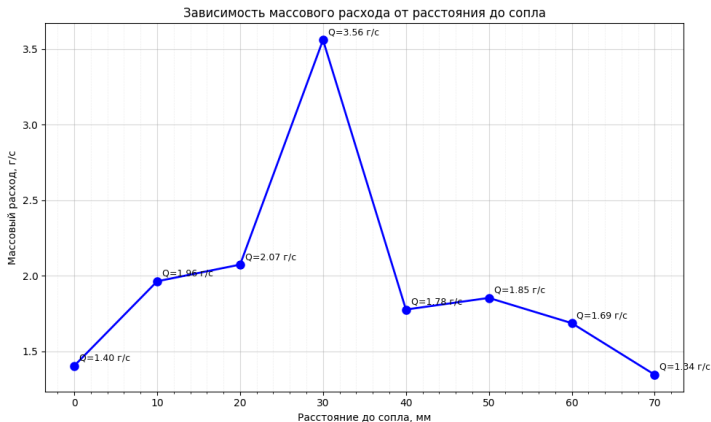
### Метод трапеций для вычисления интеграла

```
def calculate_flow_rate(positions, velocities):  
  
    positions_m = positions / 1000  
    V_times_r = velocities * np.abs(positions_m)  
    integral = 0  
    for i in range(len(positions_m) - 1):  
        f_i = V_times_r[i]  
        f_i1 = V_times_r[i + 1]  
        dr = 0.00025  
        integral += (f_i + f_i1) * dr / 2  
    volumetric_flow = 2 * np.pi * integral  
    mass_flow = volumetric_flow * rho * 1000  
  
    return mass_flow
```

# Профили скорости



# Зависимость расхода



## Результаты

- Проведена калибровка оборудования
- Построены профили скорости в различных сечениях
- Определена зависимость расхода от расстояния до сопла

## Выводы

- Теория согласуется с экспериментом до расстояний 30 мм
- На больших расстояниях влияние турбулентных потоков
- Наблюдается увеличение расхода с расстоянием в связи с диффузией