**Отчет по физике на тему:**

**Шаровой резервуар**

Выполнили:

Сухов Владимир, Нафиков Айдар

ИТМО, 2022

Теоретическая часть

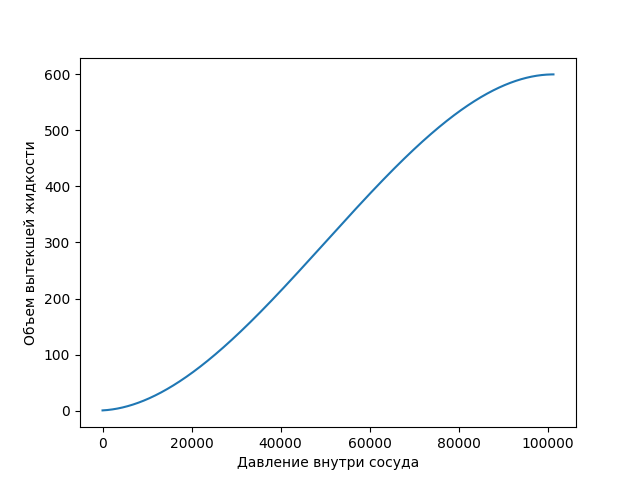
Для построения модели вычисления вытекшей жидкости из закрытого резервуара, важно вспомнить основные принципы гидростатики. Так как резервуар закрытый, после вытекания некоторого количество воды между верхней частью сосуда и плоскостью воды будет находиться вакуум, который остановит ток воды через узкое отверстие. Составим модель баланса сил, действующих на отверстие:

Отсюда находим высоту уровня оставшейся воды в резервуаре и найдя её вычисляем объем вытекшей воды.

Практическая часть

import math  
import matplotlib.pyplot as plt  
import tkinter  
import time  
  
PI = math.pi  
g = 9.8067  
Patm = 101325 # атмосферное давление  
Row = 1000 # плотность воды  
x=[]  
y=[]  
Pin = 16000 # давление внутри резервуара  
  
# По основному уравнению гидростатики, при которой происходит баланс системы  
h = (Patm-Pin)/(Row \* g)  
  
Vs = 600.0 # объем резервуара  
Dotv = 0.12 # d отверстия  
Sotv = PI\*(Dotv/2)\*\*2  
rshar = (3.0 \* Vs / (4.0 \* PI))\*\*(1/3) # Радиус шара  
  
hout = 2\*rshar - h # незаполненной части шара  
  
Vout = (PI \* hout\*\*2 ) \* (rshar - (1/3) \*hout)  
  
 # x.append(Pin)  
 # y.append(Vout)  
# print(Vout)  
#  
# plt.plot(x,y)  
# plt.xlabel("Давление внутри сосуда")  
# plt.ylabel("Объем вытекшей жидкости")  
# plt.show()  
  
  
  
Window\_Width = 800  
Window\_Height = 600  
tank\_x = 400  
tank\_y = 300  
Tank\_Radius = 100  
  
otv\_x = tank\_x  
otv\_y = tank\_y + Tank\_Radius  
otv\_R = Dotv/2/rshar\*Tank\_Radius  
Tank\_h = 200  
Tank\_point\_h = h/(2\*rshar)\*Tank\_h  
print(Tank\_point\_h)  
pointline\_r = math.sqrt(Tank\_Radius\*\*2 - abs(Tank\_point\_h - Tank\_Radius)\*\*2)  
pointline\_y = tank\_y + Tank\_Radius - Tank\_point\_h  
  
  
Ball\_min\_movement = 5  
  
Refresh\_Sec = 0.01  
  
  
def create\_animation\_window():  
 Window = tkinter.Tk()  
 Window.title("Ball Tank")  
  
 Window.geometry(f'{Window\_Width}x{Window\_Height}')  
 return Window  
  
  
def create\_animation\_canvas(Window):  
 canvas = tkinter.Canvas(Window)  
 canvas.configure(bg="White")  
 canvas.pack(fill="both", expand=True)  
 return canvas  
  
  
def animate\_ball(Window, canvas, xinc, yinc):  
 line = canvas.create\_line(tank\_x - pointline\_r,  
 pointline\_y,  
 tank\_x + Tank\_Radius + 130,  
 pointline\_y)  
 text = canvas.create\_text(tank\_x + Tank\_Radius + 130,  
 pointline\_y+10, text="Уровень воды в резервуаре",  
 font="Verdana 12",fill="black")  
 ball = canvas.create\_oval(tank\_x - Tank\_Radius,  
 tank\_y - Tank\_Radius,  
 tank\_x + Tank\_Radius,  
 tank\_y + Tank\_Radius,  
 fill="White", outline="Black", width=2)  
 otv = canvas.create\_oval(otv\_x - otv\_R,  
 otv\_y - otv\_R/2,  
 otv\_x + otv\_R,  
 otv\_y + otv\_R/2,  
 fill="Blue", outline="Black", width=1)  
  
 point\_line = canvas.create\_oval(tank\_x - pointline\_r,  
 pointline\_y - pointline\_r/4,  
 tank\_x + pointline\_r,  
 pointline\_y + pointline\_r/4,  
 fill="Blue", outline="Black", width=1)  
 while True:  
 canvas.coords(text)  
 canvas.coords(ball)  
 canvas.coords(otv)  
 canvas.coords(point\_line)  
 canvas.coords(line)  
 Window.update()  
 time.sleep(Refresh\_Sec)  
 ball\_pos = canvas.coords(ball)  
  
  
  
  
Animation\_Window = create\_animation\_window()  
Animation\_canvas = create\_animation\_canvas(Animation\_Window)  
animate\_ball(Animation\_Window, Animation\_canvas, Ball\_min\_movement, Ball\_min\_movement)

Графики и визуализация модели:



Пример визуализации при входных данных(Pin = 16000 Па)



Вывод:

Мы построили теоретическую и практическую модель решения задачи на нахождения вытекшего объема воды в закрытом шаровом резервуаре. Реализовали визуализацию и построили график зависимости объема вытекшей воды от давления внутри шарового резервуара.