Homework 1 Report

Date: January 23, 2020

To: Professor Andrew Ning

From: Joshua Canlas

The purpose of this document is to give a report on the homework assignment given to the class on Wednesday, January 15, 2020. The document will be divided into three sections (for each problem). Attached to the end of this report is an Appendix, where other documents can be found to help support the conclusions and claims made in this report.

Problem 1 (Control Volume Analysis)

Introduction
The problem asked us to determine the drag of a cylinder inside a large wind tunnel (see Figure 1). Additionally, we were to find the coefficient of drag based on the projected area and explain whether it is a reasonable answer.

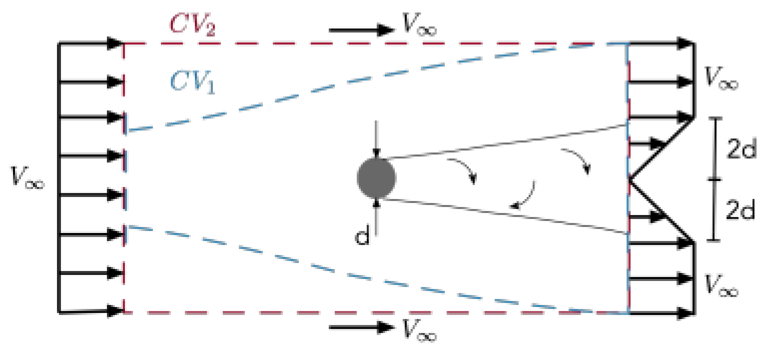


Figure 1: Sketch of velocity profiles at the inlet and outlet, the geometry of the rigid body, and the control volume. We chose control volume 1 (which is along the streamline) for this problem.

Methods and Assumptions
We chose CV_1 to be our control volume. We also made the following assumptions to help us solve for the governing equations: steady, incompressible, streamlined, and uniform velocity profile across the left control surface. The first equation we used was the continuity equation

$$0 = \frac{\partial}{\partial t}\int_{CV} \rho \, dV + \int_{CS} \rho(V \bullet dA)$$

The other equation that we used to solve for the coefficient of drag was the momentum equation

$$\sum F = \frac{\partial}{\partial t}\frac{\partial}{\partial t}\int_{CV} \rho V(V \bullet dV) + \int_{CS} \rho V(V \bullet dA)$$

Once we applied the assumptions we made, we then solved for the force of drag (see Figure A.1 for more details on the calculation)

$$D = \frac{2}{3}\rho V_\infty^2\, dL$$

where D is the force due to drag, $V_\infty$is the freestream velocity, and L is the unit length span of the airfoil. Substituting this value of drag for the equation for coefficient of drag, we found the value of the coefficient of drag to be $C_D = \frac{4}{3}$.

Results and Discussion

Because of the assumptions we made at the beginning of the problem (especially steady flow), we can assume that the value we found for the coefficient of drag is an underestimate. If the flow was unsteady, the drag would be higher, causing the coefficient of drag to increase as well. However, the value that we found for the coefficient of drag does lie in the range of values for coefficients of drag found in aerodynamic applications.

Problem 2 (CFD)

Introduction

We were asked to perform a grid convergence study for an inviscid airfoil (NACA 2412) analysis. At the end of the analysis, we produced two plots showing convergence (lift coefficient and drag coefficient) as a function of some mesh parameter (for this assignment, we chose base size).

Methods and Assumptions

The flow analysis was all done in StarCCM+. Like mentioned in the problem statement, the geometry that we used for this analysis was a NACA 2412 airfoil (points created and imported from airfoiltools.com). We followed the best practices mentioned under the tutorial guide found in Steve Portal when we created the fluid domain.

To create the mesh for the simulation, the following parameters were set:

- Polygonal
- Base size set to 1 meter (although we varied this value later in the simulation)
- Curvature set to 72 (1 point every 5 degrees)
- Surface growth rate set to 1.15
- Custom surface control at inlet and outlet (set relative size to 100 base size)*****
- Custom surface control at leading and trailing edge (set relative size to 0.1 percent of the base size with minimum size being 0.05 percent of the base size)

Furthermore, the physics model was created with the following assumptions: steady, incompressible, 2-dimensional, laminar flow, segregated flow, and constant density. The initial conditions for the wind velocity was set to 9.96 m/s in the x-direction and 0.87 m/s in the y-direction. This made it so that the angle of attack of the wing was approximately 5 degrees.

Results and Discussion
As mentioned before, we changed the base size value to find what the true values of the coefficient of lift and drag are. The mesh got better and better as we decreased the value of the base size (see Figure A.2 in the Appendix). As shown in Figure 2, the values for the coefficient of lift and drag did not vary much anymore once the base size value was 0.01 meters. Therefore, we considered the values found for the coefficients at this base size to be the true values. The

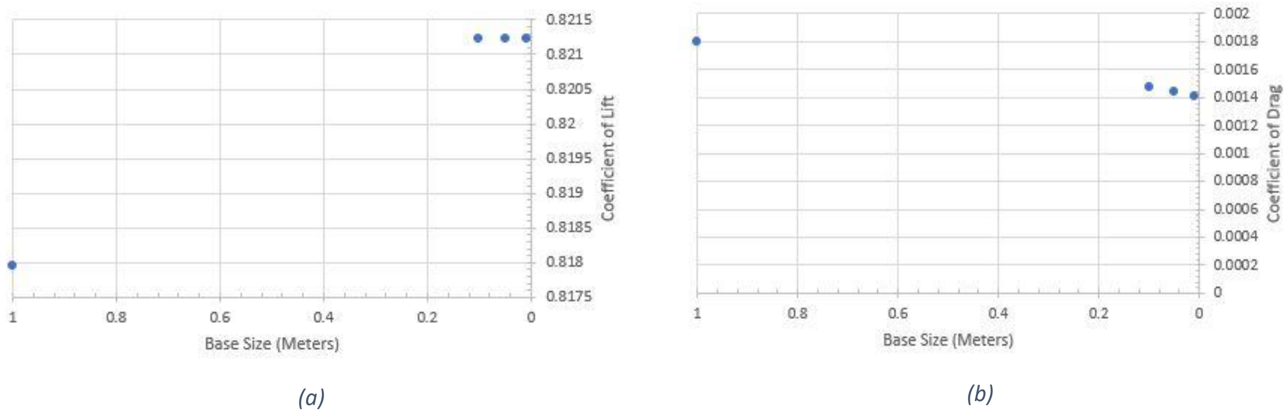


Figure 2: (a) Coefficient of lift vs base size. (b) Coefficient of drag vs base size. Notice how the values begin to come to a steady-state value when the base size is at about 0.01 meters.

values for the coefficient of lift and drag are approximately 0.82 and 0.0014 respectively. The drag coefficient is almost negligible compared to the lift coefficient. We expected the drag coefficient to be much smaller compared to the lift coefficient due to the parameters we had set for this simulation (inviscid flow). We have also provided a scalar scene from the simulation to show the velocity magnitudes at different locations of the airfoil (see Figure A.3 in the Appendix).

Problem 3 (Potential Flow)

Introduction
In this problem, we were asked to simulate leapfrogging vortices in 2D. This simulation was done in MATLAB (see Appendix for the full code). We have also included a plot of the positions of each vortex starting at 0 seconds up to 60 seconds.

Methods and Assumptions
The following parameters were used to initialize the positions and velocities of each vortex (see Figure 3):

- Vertex 1 (x,y,z)
  - Position: (0, -0.5, 0)
  - Velocity: (0, 0, 0)
- Vortex 2(x,y,z)
  - Position: (0, 0.5, 0)
  - Velocity: (0, 0, 0)
- Vortex 3 (x,y,z)
  - Position: (1, 0.5, 0)
  - Velocity: (0, 0, 0)
- Vortex 4 (x,y,z)
  - Position: (1, -0.5, 0)
  - Velocity: (0, 0, 0)

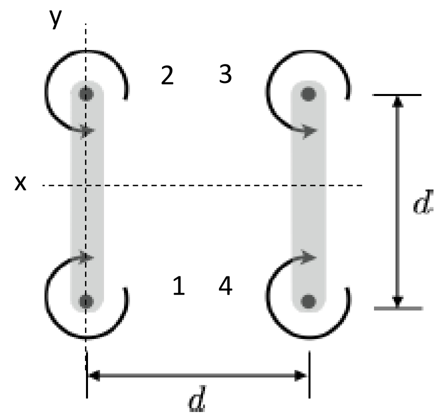


Figure 3: Initial positions and direction of rotation of each vortex. The value of d in this simulation was 1.

The overall simulation time was 40 seconds and the time step used was 0.01 seconds to provide stability.

The general equation we used to solve for each vortex's velocity at each time step is

$$v_\theta = \frac{\Gamma \times \vec{r}}{2\pi r^2}$$

where v_θ is the velocity induced by a vortex, Γ is the strength of a vortex, $\vec{r}$ is the position of a vortex relative to another vortex, and r is the magnitude of the distance of a vortex from another vortex. The induced velocities were then used to solve for the actual velocity of each vortex. To find the position of the vortex at each time step, we simply multiplied the velocity by the size of the time step (please see the Appendix for more details on how this was implemented in code).

Results and Discussion

Figure 4 shows the positions of each vortex from 0 to 60 seconds. This plot is what we expected to as a result of this simulation. It is clear that the vortices are "leapfrogging" as one pair of vortices goes past the other.

This phenomenon is all due to the velocities induced by each vortex on all of the other vortices. For example, at the beginning of the simulation, each vortex was located at initial positions with no initial velocity. Once we began the simulation, the velocity induced on vortex 1 from vortices 2, 3, and 4 all contributed to the overall velocity of vortex 1 (the same goes for the other 3 vortices). Since we defined the circulation of vortices 1 and 4 to be positive and 2 and 3 to be negative, all of the vortices

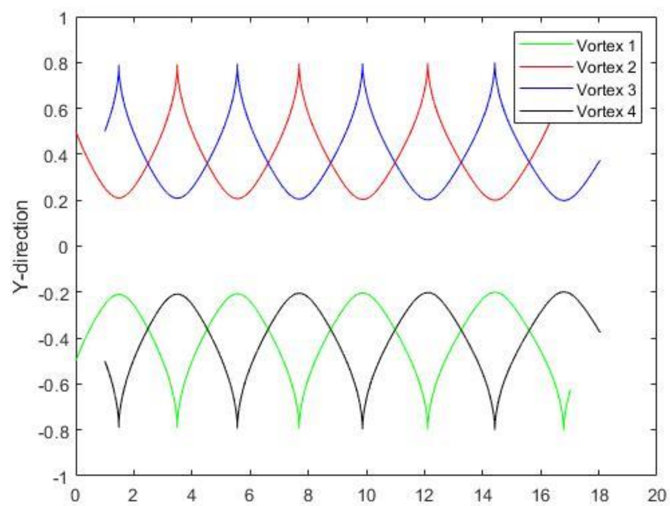


Figure 4: This plot shows the x and y positions of each vortex from 0 to 60 seconds.

moved in the positive x-direction. Vortices 3 and 4 caused 1 and 2 to come closer together in the y-direction, while 1 and 2 caused 3 and 4 to move away from each other. Over time, the positions of these vortices switched back and forth, causing the “leapfrogging” motion.

Appendix

Handwritten work (Problem 1)

Problem 1 | ME EN 512 | Joshua Carling

Assumptions:
① Steady
② Incompressible
③ Streamline
④ Uniform velocity profile across CS_1.

L goes into the page

Mass:

$$0 = \frac{\partial}{\partial t}\int_{CV} \rho\, dV + \int_{CS} \rho(\vec{V}\cdot d\vec{A})$$

$$0 = \int_{CS_1} \rho_1 \vec{V}_1\cdot d\vec{A}_1 + \int_{CS_2} \rho_2 \vec{V}_2\cdot d\vec{A}_2$$

$$0 = -\int_a^b \rho_1 \vec{V}_1\, dy\, L + \int_d^c \rho_2 \vec{V}_2\, dy\, L$$

$$\int_a^b \rho_1 \vec{V}_1\, dy = \int_d^c \rho_2 \vec{V}_2\, dy \xrightarrow{\text{Multiply both sides by } \vec{V}_1} \int_a^b \rho_1 \vec{V}_1^2\, dy = \int_d^c \rho_2 \vec{V}_1 \vec{V}_2\, dy$$

Momentum:

D' = drag per unit span, $D' = \frac{D}{L}$

$$\Sigma F = \cancel{\frac{\partial}{\partial t}\int_{CV} \rho \vec{V}}(\vec{V}\cdot d\vec{A}) + \int_{CS} \rho \vec{V}(\vec{V}\cdot d\vec{A})$$

$$-D' = -\int_a^b \rho_1 \vec{V}_1^2\, dy + \int_d^c \rho_2 \vec{V}_2^2\, dy$$

$$-D' = -\int_d^c \rho_2 \vec{V}_1 \vec{V}_2\, dy + \int_d^c \rho_2 \vec{V}_2^2\, dy$$

$$D' = \rho \int_d^c V_2 (V_1 - V_2)\, dy$$

$$V(y) = \frac{y V_\infty}{2d}$$

$$D' = \rho\int_e^c \cancel{V_\infty (V_\infty - V_\infty)}\, dy + 2\rho\int_g^e V(y)(V_\infty - V(y))\, dy + \rho\int_d^f \cancel{V_\infty (V_\infty - V_\infty)}$$

$$D' = 2\rho\int_0^{2d} \frac{y V_\infty}{2d}\left(V_\infty - \frac{y V_\infty}{2d}\right) dy = 2\rho V_\infty^2 \int_0^{2d} \frac{y}{2d} - \frac{y^2}{4d^2}\, dy$$

$$D' = 2\rho V_\infty^2 \left[\frac{4d^2}{4d} - \frac{8d^3}{12d^2}\right] = \rho V_\infty^2 d\left(2 - \frac{4}{3}\right) = \frac{2}{3}\rho V_\infty^2 d$$

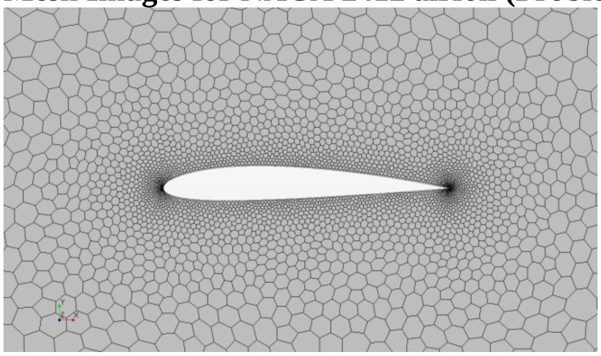
$$D = \frac{2}{3}\rho V_\infty^2 d L$$

Answer: $C_D = \frac{D}{\frac{1}{2}\rho V^2 d L} = \frac{\frac{2}{3}\rho V^2 d L}{\frac{1}{2}\rho V^2 d L}$

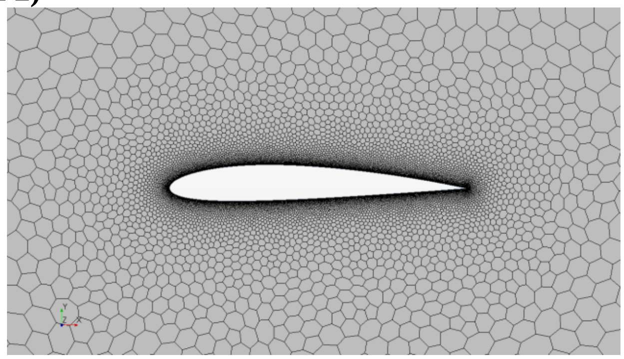
$$\boxed{C_D = 4/3}$$

Figure A.1: This is the handwritten solution for Problem 1.

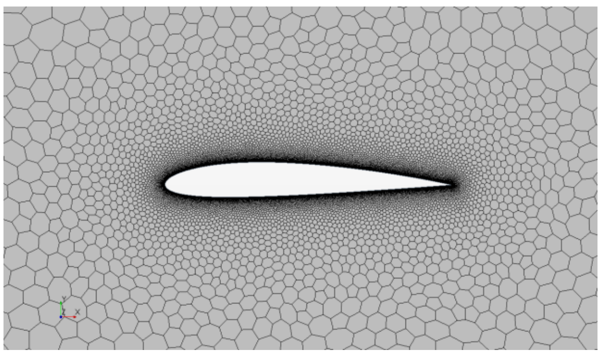
Mesh Images for NACA 2412 airfoil (Problem 2)



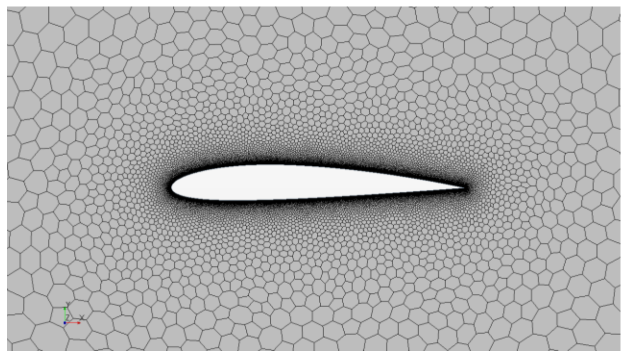
(a)



(b)



(c)



(d)

Figure A.2: Base size (a) 1, (b) 0.1, (c) 0.05, (d) 0.01

Scalar Scene (Problem 2)

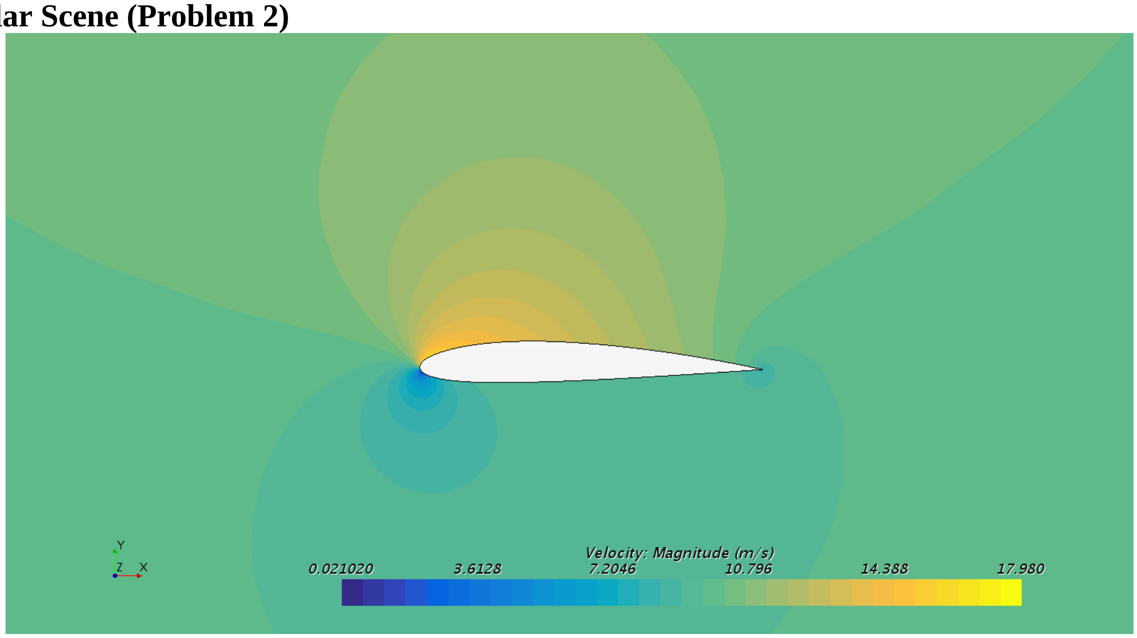


Figure A.3: Scalar scene image taken from StarCCM+. This image shows the velocity magnitude at different points on the airfoil. Notice that the highest velocity is on the top half of the airfoil and the the lowest velocities are on the bottom.

MATLAB code for "leapfrogging vortices" simulation (Problem 3)

```
clc
clear all

% Vortex 1-4 (starting at bottom left moving clockwise)
d = 1.0;
delta_t = 0.01;
t_init = 0.0;
t_final = 60.0;

gamma1 = [0, 0, -1]; % same strength for each vortex
gamma2 = [0, 0, 1];
gamma3 = [0, 0, 1];
gamma4 = [0, 0, -1];

pos_1_old = [0, -0.5, 0]; % initialize positions of vortices
pos_2_old = [0, 0.5, 0];
pos_3_old = [1.0, 0.5, 0];
pos_4_old = [1.0, -0.5, 0];

vel_1_old = [0, 0, 0]; % initialize velocities
vel_2_old = [0, 0, 0];
vel_3_old = [0, 0, 0];
vel_4_old = [0, 0, 0];

vortex_1 = zeros(4000,2); % creates empty matrices
vortex_2 = zeros(4000,2);
vortex_3 = zeros(4000,2);
vortex_4 = zeros(4000,2);

count = 1;

for R = t_init:delta_t:t_final
    % computes the x and y distances between each vortex
    dist_12_x = pos_2_old(1)-pos_1_old(1);
    dist_13_x = pos_3_old(1)-pos_1_old(1);
    dist_14_x = pos_4_old(1)-pos_1_old(1);
    dist_21_x = pos_1_old(1)-pos_2_old(1);
    dist_23_x = pos_3_old(1)-pos_2_old(1);
    dist_24_x = pos_4_old(1)-pos_2_old(1);
    dist_31_x = pos_1_old(1)-pos_3_old(1);
    dist_32_x = pos_2_old(1)-pos_3_old(1);
    dist_34_x = pos_4_old(1)-pos_3_old(1);
    dist_41_x = pos_1_old(1)-pos_4_old(1);
    dist_42_x = pos_2_old(1)-pos_4_old(1);
    dist_43_x = pos_3_old(1)-pos_4_old(1);
    dist_12_y = pos_2_old(2)-pos_1_old(2);
    dist_13_y = pos_3_old(2)-pos_1_old(2);
    dist_14_y = pos_4_old(2)-pos_1_old(2);
    dist_21_y = pos_1_old(2)-pos_2_old(2);
    dist_23_y = pos_3_old(2)-pos_2_old(2);
    dist_24_y = pos_4_old(2)-pos_2_old(2);
    dist_31_y = pos_1_old(2)-pos_3_old(2);
    dist_32_y = pos_2_old(2)-pos_3_old(2);
```

```
    dist_34_y = pos_4_old(2)-pos_3_old(2);
    dist_41_y = pos_1_old(2)-pos_4_old(2);
    dist_42_y = pos_2_old(2)-pos_4_old(2);
    dist_43_y = pos_3_old(2)-pos_4_old(2);

    % computes the induced velocity of each vortex
    vel_12 = cross(gamma1,
[dist_12_x,dist_12_y,0])/(2*pi*(dist_12_x^2+dist_12_y^2));
    vel_13 = cross(gamma1,
[dist_13_x,dist_13_y,0])/(2*pi*(dist_13_x^2+dist_13_y^2));
    vel_14 = cross(gamma1,
[dist_14_x,dist_14_y,0])/(2*pi*(dist_14_x^2+dist_14_y^2));
    vel_21 = cross(gamma2,
[dist_21_x,dist_21_y,0])/(2*pi*(dist_21_x^2+dist_21_y^2));
    vel_23 = cross(gamma2,
[dist_23_x,dist_23_y,0])/(2*pi*(dist_23_x^2+dist_23_y^2));
    vel_24 = cross(gamma2,
[dist_24_x,dist_24_y,0])/(2*pi*(dist_24_x^2+dist_24_y^2));
    vel_31 = cross(gamma3,
[dist_31_x,dist_31_y,0])/(2*pi*(dist_31_x^2+dist_31_y^2));
    vel_32 = cross(gamma3,
[dist_32_x,dist_32_y,0])/(2*pi*(dist_32_x^2+dist_32_y^2));
    vel_34 = cross(gamma3,
[dist_34_x,dist_34_y,0])/(2*pi*(dist_34_x^2+dist_34_y^2));
    vel_41 = cross(gamma4,
[dist_41_x,dist_41_y,0])/(2*pi*(dist_41_x^2+dist_41_y^2));
    vel_42 = cross(gamma4,
[dist_42_x,dist_42_y,0])/(2*pi*(dist_42_x^2+dist_42_y^2));
    vel_43 = cross(gamma4,
[dist_43_x,dist_43_y,0])/(2*pi*(dist_43_x^2+dist_43_y^2));

    if R == 0
        vel_1_new = [0,0,0];
        vel_2_new = [0,0,0];
        vel_3_new = [0,0,0];
        vel_4_new = [0,0,0];
    else
        % updates the velocities of each vortex
        vel_1_new = vel_21 + vel_31 + vel_41;
        vel_2_new = vel_12 + vel_32 + vel_42;
        vel_3_new = vel_13 + vel_23 + vel_43;
        vel_4_new = vel_14 + vel_24 + vel_34;
    end

    vel_1_old = vel_1_new;
    vel_2_old = vel_2_new;
    vel_3_old = vel_3_new;
    vel_4_old = vel_4_new;

    % updates the position of each vortex
    pos_1_new = pos_1_old + (vel_1_old * 0.01);
    pos_2_new = pos_2_old + (vel_2_old * 0.01);
    pos_3_new = pos_3_old + (vel_3_old * 0.01);
    pos_4_new = pos_4_old + (vel_4_old * 0.01);

    pos_1_old = pos_1_new;
```

```
    pos_2_old = pos_2_new;
    pos_3_old = pos_3_new;
    pos_4_old = pos_4_new;

    % saves the position in its respective matrix (for plotting purposes)
    vortex_1(count,:) = [pos_1_old(1), pos_1_old(2)];
    vortex_2(count,:) = [pos_2_old(1), pos_2_old(2)];
    vortex_3(count,:) = [pos_3_old(1), pos_3_old(2)];
    vortex_4(count,:) = [pos_4_old(1), pos_4_old(2)];

    count = count + 1;
end

% creates plot for the simulation
figure(1)
plot(vortex_1(:,1), vortex_1(:,2), 'g')
hold on
plot(vortex_2(:,1), vortex_2(:,2), 'r')
plot(vortex_3(:,1), vortex_3(:,2), 'b')
plot(vortex_4(:,1), vortex_4(:,2), 'k')
legend('Vortex 1', 'Vortex 2', 'Vortex 3', 'Vortex 4')
xlabel('X-direction')
ylabel('Y-direction')
```