Numerical Simulation of Thrust for a Flapping and Morphing Airfoil

Ishaan Sharma (22AE10013)
Kalavakunta Kalyani (22AE10017)
Bhandaru Uma Nethrika (22AE10007)

Low Speed Aerodynamics

Spring 2024

Contents

1 Abstract

In this paper, we aim to perform numerical analyses on morphing Joukowski airfoils undergoing flapping and heaving motions using ANSYS Fluent. Potential flow solutions offer valuable insights into the aerodynamics of fluid flow. Using the Joukowski transformation, we map the circular cylinder to an airfoil, helping us find potential flow solutions for complex airfoil shapes. By comparing the resulting aerodynamic properties, particularly lift and drag, with theoretical predictions, our paper aims to validate and improve our understanding of aerodynamic performance of morphing airfoils.

2 Introduction

Numerous studies have investigated the aerodynamic performance of bio-inspired airfoils. In observing aquatic creatures' remarkable ability to propel themselves forward through water by dynamically morphing their bodies and birds' adeptness at generating lift through wing pitching, researchers have recognized the potential for enhancing aerodynamic efficiency through bio-inspired design. Aquatic creatures, such as fish, have evolved sophisticated mechanisms for propulsion by dynamically altering the shape of their bodies. Through subtle adjustments in fin shape and flexibility, fishes can efficiently navigate complex hydrodynamic environments. Similarly, birds employ wing morphing strategies to optimize lift generation and maneuverability during flight. These observations emphasize the effectiveness of morphing structures in achieving superior aerodynamic performance across diverse environments. Building upon this biological inspiration, our paper focuses on the design and development of bio-inspired airfoils capable of morphing. Using the Joukowski transformation as a mathematical tool we seek to create airfoils that can morph their shapes to optimize lift and drag characteristics.

3 Theoretical Formulation

3.1 Joukowski Transformation

Joukowski airfoils, attributed to Nikolai Zhukovsky, arise from the application of the Joukowski transform, a mathematical technique developed to map the flow dynamics around a circular cylinder onto those around an airfoil contour. By implementing this transformation on a circular geometry within the complex plane, a range of airfoil profiles can be generated. These resultant airfoils commonly exhibit streamlined configurations, facilitating lift generation under aerodynamic loading conditions.

The transformation used is shown below :

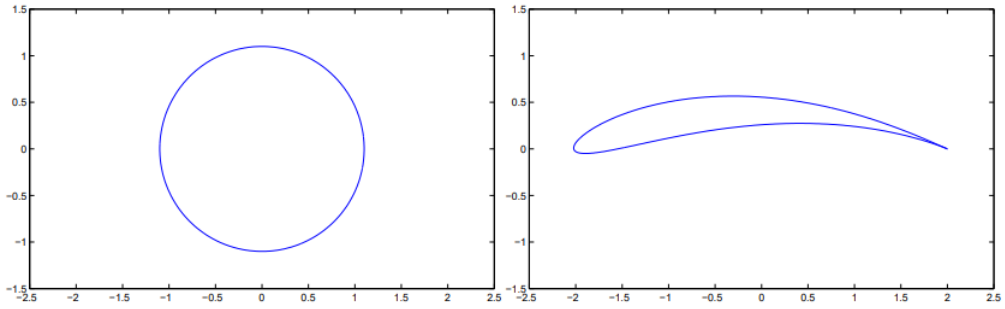


Figure 1: Joukowski mapping of an airfoil from cylinder

$$z = \zeta + \zeta_c + \frac{a^2}{\zeta + \zeta_c} \tag{1}$$

where

$$\zeta_c = \zeta_x + i\zeta_y$$

The size and shape of the foil are determined by the parameters a and ζ_c, and the radius of C is $r_c = ||a - \zeta_c||$. The real number a always has the opposite sign of the real part of ζ_c.

By changing the parameter ζ_y we can generate morphing airfoils. So we choose $\zeta_x = -0.0625$ and $\zeta_y = -0.01sin(\pi t)$.
The parameter a is set to 0.05 and r_c is set to 0.06.

The inverse of the Joukowski map is given by:

$$\zeta = F^{-1}(z) = \frac{1}{2}\left(z + \sqrt{z^2 - 4a^2}\right) - \zeta_c$$

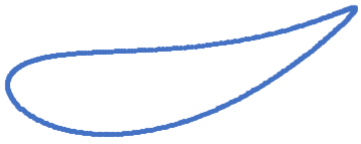


Figure 2: Morphed airfoil when $\zeta_y = -0.01$ viewed at an angle $-10°$

Figure 3: Morphed airfoil when $\zeta_y = 0$

Figure 4: Morphed airfoil when $\zeta_y = 0.01$ viewed at an angle 10°

3.2 Velocity Potential

Let us assume the fluid is irrotational and incompressible. So we can represent the potential function and stream function using complex potential notation.

$$w(z) = \phi(x, y) + i\psi(x, y) \tag{2}$$

This complex potential can be decomposed into contributions from motion of the airfoil (w_1, w_2, w_3) and shape deformations(w_1^s, w_2^s, w_3^s), allowing for a superposition of potentials to describe the flow field efficiently[3].

$$w(z) = Uw_1(\zeta) + Vw_2(\zeta) + \Omega w_3(\zeta) + \dot{\zeta}_x w_1^s(\zeta) + \dot{\zeta}_y w_2^s(\zeta) + aw_3^s(\zeta) \tag{3}$$

where

$$w_1 = -\frac{r_c^2}{\zeta} + \zeta_c + \frac{a^2}{\zeta + \zeta_c}$$
$$w_2 = \frac{-ir_c^2}{\zeta} - i\zeta_c + \frac{-ia^2}{\zeta + \zeta_c}$$
$$w_3 = \frac{-i}{2}[r_c^2 + 2\zeta_c \frac{r_c^2}{\zeta} + \delta^2 + 2a^2 \frac{\frac{r_c^2}{\zeta} + \overline{\zeta_c}}{\zeta + \zeta_c} + \frac{a^4(\zeta - \zeta_c)}{(\zeta - \zeta_c)(r_c^2 - \delta^2)}]$$

$$w_1^s = \frac{-r_c^2}{\zeta} + \frac{a^2 r_c^2}{\zeta\zeta_c} + \frac{a^2}{(\zeta+\zeta_c)} - \frac{a^4\zeta_c^2}{(\zeta+\zeta_c)(r_c^2-\delta^2)^2} - \frac{2a^4 r_c^2\zeta_c}{(r_c^2-\delta^2)^3}$$
$$\log\left(\frac{\zeta+\zeta_c}{r_c}\right) - \left(\frac{a^2 r_c^2}{\zeta_c^3} - \frac{a^4 r_c^2}{\zeta_c(r_c^2-\delta^2)^2}\right)\frac{\zeta}{(\zeta+\zeta_c)}$$
$$- 2a^2 r_c^2\left(\frac{1}{\zeta_c^3} + \frac{\overline{\zeta_c}a^2}{(r_c^2-\delta^2)^3}\right)\log\left(\frac{\zeta+\zeta_c}{\zeta}\right) + \frac{2a^4 r_c^2(i\zeta_y)}{(r_c^2-\delta^2)^3}\log\left(\frac{\zeta}{r_c}\right)$$

$$w_2^s = (-i)\left(\frac{r_c^2}{\zeta} - \frac{a^2 r_c^2}{\zeta\zeta_c^2} + \frac{a^2}{(\zeta+\zeta_c)} - \frac{a^2\zeta_c^2}{(\zeta+\zeta_c)(r_c^2-\delta^2)^2}\right) - \frac{2a^4 r_c^2\zeta_c\log\left(\frac{(\zeta+\zeta_c)}{r_c}\right)}{(r_c^2-\delta^2)^3}$$
$$+ \left(\left(\frac{a^2 r_c^2}{\zeta_c^3} - \frac{a^4 r_c^2}{\zeta_c(r_c^2-\delta^2)^2}\right)\frac{\zeta}{(\zeta+\zeta_c)} + 2a^2 r_c^2\left(\frac{1}{\zeta_c^3} + \frac{\overline{\zeta_c}a^2}{(r_c^2-\delta^2)^3}\right)\log\left(\frac{\zeta+\zeta_c}{\zeta}\right) + \frac{2a^4 r_c^2\zeta_x}{(r_c^2-\delta^2)^3}\log\left(\frac{\zeta}{r_c}\right)\right)$$

$$w_3^s = 2a\left(\frac{-r_c^2}{\zeta\zeta_c} - \frac{a^2\zeta_c}{(\zeta+\zeta_c)(r_c^2-\delta^2)} - \frac{a^2 r_c^2\log((\zeta+\zeta_c)/r_c)}{(r_c^2-\delta^2)^2} + \frac{r_c^2((r_c^2-\delta^2)^2 - a^2\zeta_c^2)}{\zeta_c^2(r_c^2-\delta^2)^2}\log\left(\frac{(\zeta+\zeta_c)}{\zeta}\right)\right)$$

3.3 Forces Acting on the Airfoil

In an inertial frame instantaneously aligned with the z coordinate frame, in which the fluid at infinity is at rest, the pressure p at each point in the fluid is given by

$$\begin{aligned} p &= -\frac{\partial\phi}{\partial t} - \frac{1}{2}(u^2+v^2) \\ &= -\left(\frac{\partial}{\partial t}\frac{w+\bar{w}}{2}\right) - \frac{1}{2}\left(\frac{dw}{dz}\frac{\bar{dw}}{dz}\right) \end{aligned}$$

Using X and Y to represent the forces on the foil along the x- and y-axes respectively, we see that

$$X = \int_\Sigma p(-dy)$$
$$Y = \int_\Sigma p\ dx$$

or equivalently

$$X + iY = i\int_\Sigma p\ dz$$

So,

$$X + iY = -\frac{i}{2}\int_\Sigma\left(\frac{\partial w}{\partial t} + \frac{\partial\bar{w}}{\partial t} + \frac{dw}{dz}\frac{\bar{dw}}{dz}\right)$$

Finally, after evaluating the integrals for the velocity potential w we defined earlier and assuming $U = 0$, we get the equation for forces acting on the airfoil [3].

$$X + iY = -\dot{V}A_2 - \dot{\Omega}A_3 + \ddot{\zeta_y}B_2 + V^2C_3 + V\Omega C_5 + \Omega^2 C_6 + V\dot{\zeta_y}D_6$$
$$+ V\dot{r_c}D_8 + \Omega\dot{\zeta_y}D_{10} + \Omega\dot{r_c}D_{12} + \dot{\zeta_y}^2 E_3 + \dot{\zeta_y}\dot{r_c}E_8$$

where

$$A_2 = i\pi\left(2a^2 + r_c^2 + \frac{a^4 r_c^2}{(r_c^2-\delta^2)^2)}\right)$$

$$A_3 = -\left(\frac{i\pi\left(a^6 r_c^2\bar{\zeta_c} + 2a^4\zeta_c(r_c^2-\delta^2)^2 - r_c^2\zeta_c(r_c^2-\delta^2)^3 + 2a^2\bar{\zeta_c}(-r_c^2+\delta^2)^3\right)}{(r_c^2-\delta^2)^3}\right)$$

$$\begin{aligned}B_2 = \frac{\pi}{(r_c^2-\delta^2)^4}\big[&4a^5 r_c^2(i\zeta_c - \bar{\zeta_c})(r_c^2-\delta^2) - 2a^2 i(r_c^2-\delta^2)^4 - ir_c^2(r_c^2-\delta^2)^4 - \\ &2a^4(r_c^2-\delta^2)(-(r_c^2\delta^2) + i(r_c^4 - r_c^2\zeta_c^2 + \zeta_c^3\bar{\zeta_c})) \\ &- a^6 r_c^2(2(\bar{\zeta_c})^2 + i(r_c^2 + (2\zeta_c - \bar{\zeta_c})\bar{\zeta_c}))]\end{aligned}$$

$$C_3 = \frac{-2a^2\pi\zeta_c(r_c^2-\delta^2)((r_c^2-\delta^2)^3 + a^2(r_c^4 + \zeta_c^3\bar{\zeta_c}))}{(r_c^2(r_c^2+\zeta_c(a-\bar{\zeta_c}))^2(r_c^2-\zeta_c(a+\bar{\zeta_c}))^2)}$$

$$\begin{aligned}C_5 = {}&2a^2\pi((r_c^2-2\delta^2)(r_c^2-\delta^2)^4 \\ &+ a^2(r_c^8 + r_c^6\zeta_c(3\zeta_c - 2\bar{\zeta_c}) + 4r_c^2\zeta_c^4\bar{\zeta_c}^2 + r_c^4\zeta_c^2\bar{\zeta_c}(-7\zeta_c + \bar{\zeta_c})) \\ &+ a^4\zeta_c(-(r_c^2\zeta_c^3) + 2\zeta_c^4\bar{\zeta_c} + r_c^4(\zeta_c + \bar{\zeta_c}))) \\ &/(r_c^2(r_c^2+\zeta_c(a-\bar{\zeta_c}))^2(r_c^2-\zeta_c(a+\bar{\zeta_c}))^2)\end{aligned}$$

$$\begin{aligned}C_6 = {}&-2a^2\pi(-(\bar{\zeta_c}(r_c^2-\delta^2)^6) + a^6(-1(r_c^2\zeta_c^5) + r_c^4\zeta_c^2\bar{\zeta_c} + \zeta_c^6\bar{\zeta_c}) \\ &+ a^2\zeta_c(r_c^2-\delta^2)^3(2r_c^4 - 4r_c^2\delta^2 + \zeta_c^2\bar{\zeta_c}^2) \\ &+ a^4(r_c^8\bar{\zeta_c} + \zeta_c^6\bar{\zeta_c}^3 + r_c^6(\zeta_c^3 - 2\zeta_c\bar{\zeta_c}^2) + r_c^4(-2\zeta_c^4\bar{\zeta_c} + \zeta_c^2\bar{\zeta_c}^3))) \\ &/(r_c^2(r_c^2+\zeta_c(a-\bar{\zeta_c}))^2(r_c^2-\delta^2)(r_c^2-\zeta_c(a+\bar{\zeta_c}))^2)\end{aligned}$$

$$\begin{aligned}D_6 = {}&-2a^2\pi(2\zeta_c(r_c^2-\delta^2)^7 + a^2(r_c^2-\delta^2)^4(-4r_c^2\zeta_c^3 + 4\zeta_c^4\bar{\zeta_c} \\ &+ r_c^4(3\zeta_c - 2\bar{\zeta_c} - i\zeta_y)) + a^6 r_c^2\zeta_c^2(\zeta_c^2\bar{\zeta_c}^2(\zeta_c - i\zeta_y) \\ &+ r_c^2\zeta_c^2(2\zeta_c - 2\bar{\zeta_c} - i\zeta_y) + r_c^4(-\bar{\zeta_c} - i\zeta_y)) + a^4(r_c^2-\delta^2)^2(-2\zeta_c^6\bar{\zeta_c} \\ &+ r_c^6(\zeta_c - \bar{\zeta_c} - i\zeta_y) - r_c^4\zeta_c(4\zeta_c^2 - 6\delta^2 + \bar{\zeta_c}^2 - 2i\zeta_c\zeta_y - 2i\bar{\zeta_c}\zeta_y) \\ &+ r_c^2\zeta_c^2(2\zeta_c^3 - \zeta_c\bar{\zeta_c}^2 - i\bar{\zeta_c}^2\zeta_y))) \\ &/(r_c^2(r_c^2+\zeta_c(a-\bar{\zeta_c}))^2(r_c^2-\delta^2)^3(r_c^2-\zeta_c(a+\bar{\zeta_c}))^2)\end{aligned}$$

$$D_8 = \frac{-2i\pi(a^2+r_c^2)}{r_c}$$

$$\begin{aligned}D_{10} = {}&[\pi r_c^4(r_c^2-\delta^2)^8 + a^2(r_c^2-\delta^2)^6(-2\pi r_c^4\zeta_c^2 + \pi(r_c^2-4\delta^2)(r_c^2-\delta^2)^2) \\ &+ a^1 0\pi r_c^2\zeta_c^4(r_c^4 + 2\zeta_c\bar{\zeta_c}^2(\zeta_c - i\zeta_y) + 2r_c^2\bar{\zeta_c}(2\zeta_c - 2\bar{\zeta_c} - i\zeta_y)) \\ &- a^4(r_c^2-\delta^2)^4(\pi r_c^4(r_c^4 - \zeta_c^4 - 2r_c^2\delta^2 + \zeta_c^2\bar{\zeta_c}^2) \\ &- \pi\zeta_c(-r_c^2+\delta^2)(r_c^2\zeta_c^2\bar{\zeta_c} + 4\zeta_c^3\bar{\zeta_c}^2 + r_c^4(-5\zeta_c - 2\bar{\zeta_c} - 2i\zeta_y))) \\ &+ a^6(r_c^2-\delta^2)^3(2\pi r_c^4\zeta_c^2(r_c^2-\delta^2) + \pi(3r_c^8 - 4\zeta_c^6\bar{\zeta_c}^2 \\ &+ r_c^2\zeta_c^2\bar{\zeta_c}(17\zeta_c^3 + 2\zeta_c\bar{\zeta_c}^2 + 2i\bar{\zeta_c}^2\zeta_y) + r_c^6(6\zeta_c^2 + 5\delta^2 - 2\bar{\zeta_c}(3\bar{\zeta_c} - i\zeta_y)) \\ &+ r_c^4\zeta_c(-13\zeta_c^3 - 10\zeta_c\bar{\zeta_c}^2 + 2\bar{\zeta_c}^2(3\bar{\zeta_c} - 2i\zeta_y) + \zeta_c^2(-10\bar{\zeta_c} - 4i\zeta_y)))) \\ &- a^8\zeta_c(r_c^2-\delta^2)(\pi r_c^4\zeta_c^3(r_c^2-\delta^2) + \pi(11r_c^2\zeta_c^6\bar{\zeta_c} - 4\zeta_c^7\bar{\zeta_c}^2 \\ &+ 8r_c^8(\bar{\zeta_c} - i\zeta_y) - r_c^4\zeta_c^2(7\zeta_c^3 + 4\zeta_c\bar{\zeta_c}^2 - 6\bar{\zeta_c}^3 - 8i\bar{\zeta_c}^2\zeta_y + \zeta_c^2(4\bar{\zeta_c} + 2i\zeta_y)) \\ &+ 2r_c^6\zeta_c(\zeta_c^2 + 2\delta^2 + \bar{\zeta_c}(-7\bar{\zeta_c} - 8i\zeta_y))))] \\ &/(r_c^2(r_c^2+\zeta_c(a-\bar{\zeta_c}))^2(r_c^2-\delta^2)^4(r_c^2-\zeta_c(a+\bar{\zeta_c}))^2)\end{aligned}$$

$$
\begin{aligned}
D_{12} &= \frac{-2i}{r_c(r_c^2-\delta^2)^4}(a^6\pi r_c^4\bar{\zeta_c} + \pi r_c^2\zeta_c(r_c^2-\delta^2)^4 + a^2\pi\bar{\zeta_c}(r_c^2-\delta^2)^4 \\
&\quad + a^4\zeta_c(r_c^2-\delta^2)^2(\pi r_c^2 - 2\pi r_c^2 + \pi\delta^2)) \\
E_3 &= -2a^2\pi(\zeta_c(r_c^2-\delta^2)^5 - 2a^3r_c^4(r_c^2-\delta^2)((-1+i)r_c^2 + 3i\bar{\zeta_c}\zeta_y \\
&\quad + \zeta_c(2(-1+i)\bar{\zeta_c} + 3i\zeta_y)) + a^4r_c^2(\zeta_c^2{\bar{\zeta_c}}^2(-\zeta_c - i\zeta_y) \\
&\quad + r_c^4(4\zeta_c + (-4-i)\bar{\zeta_c} - i\zeta_y) + r_c^2\bar{\zeta_c}(5\zeta_c^2 + 2{\bar{\zeta_c}}^2 + \zeta_c((-3-2i)\bar{\zeta_c} + 2i\zeta_y))) \\
&\quad + a^2(r_c^2-\delta^2)(4r_c^2\zeta_c^4\bar{\zeta_c} - \zeta_c^5{\bar{\zeta_c}}^2 + r_c^6((5+i)\zeta_c - 4\bar{\zeta_c} - i\zeta_y) + r_c^4\zeta_c(-3\zeta_c^2 + \zeta_c(-5\bar{\zeta_c} + 2i\bar{\zeta_c} \\
&\quad + 3i\zeta_y) + \bar{\zeta_c}(\bar{\zeta_c} + 3i\zeta_y - i\zeta_y)))) \\
&\quad /(r_c^2(r_c^2-\delta^2)^5) \\
E_8 &= 2\pi(-(a^2(r_c^2 - \zeta_c\bar{\zeta_c})^5) - r_c^2(r_c^2 - \zeta_c\bar{\zeta_c})^5 \\
&\quad - 4a^5r_c^2(\zeta_c + i\bar{\zeta_c})(r_c^2 - \zeta_c\bar{\zeta_c})(2r_c^2 + \zeta_c\bar{\zeta_c}) \\
&\quad + a^4(r_c^2 - \zeta_c\bar{\zeta_c})(r_c^6 + (1+2i)r_c^2\zeta_c^2{\bar{\zeta_c}}^2 + \zeta_c^4{\bar{\zeta_c}}^4 + r_c^4\zeta_c(-\zeta_c + (4+4i)\bar{\zeta_c})) \\
&\quad + a^6r_c^2(r_c^4 + 2\zeta_c(\zeta_c - i\bar{\zeta_c}){\bar{\zeta_c}}^2 + r_c^2(5\zeta_c\bar{\zeta_c} - (2+4i){\bar{\zeta_c}}^2))) \\
&\quad /(r_c(r_c^2 - \zeta_c\bar{\zeta_c})^5)
\end{aligned}
$$

In the next sections we will compare the forces obtained theoretically with CFD simulation.

4 Simulation Setup

We consider a Joukowski airfoil with the following parameters for the transformation.

$$
\begin{aligned}
a &= 0.05 \\
R &= 0.06 \\
\zeta_x &= -0.00625
\end{aligned}
$$

The y-coordinate of the center of the circle being mapped to the Joukowski airfoil, varies with time as

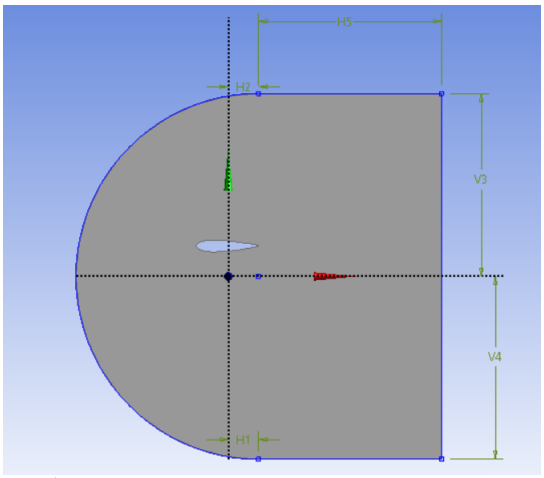
$$\zeta_y = -0.01\sin(\pi t)$$

This governs the morphing of the airfoil. Additionally, the airfoil also undergoes pitching and heaving about the 25% chord length point as

$$
\begin{aligned}
h &= 0.1\cos(\pi t) \\
\alpha &= 10 \times \left(\frac{\pi}{180}\right)\sin(\pi t)
\end{aligned}
$$

Because we are only concerned with the thrust generated by a flapping and morphing wing, we are considering that the freestream velocity is zero.

The airfoil coordinates for $t = 0$ were generated and used for initializing the geometry in ANSYS Fluent. The geometry and dimensions of the flow domain are shown in Figure 5.



Dimensions: 5	
H1	10.026 cm
H2	10.026 cm
H5	60 cm
V3	60 cm
V4	60 cm

Figure 5: Geometry and dimensions of the flow domain

A structured mesh with quadrilateral cells was created which is shown in the vicinity of the airfoil in Figure 6. In order to simulate the flapping and morphing

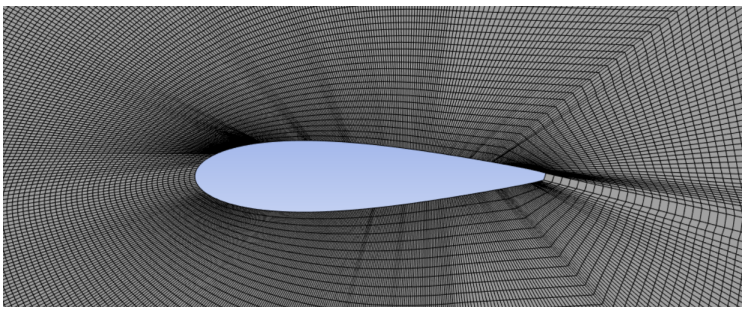


Figure 6: Mesh Generation

motion of the airfoil, we need to dynamically update the node positions of each mesh node at the airfoil surface at each timestep. This was done using the Dynamic Meshing feature in ANSYS Fluent which allows you to use user-defined motion functions. In order to update the node positions at each timestep, a user-defined function was written in C programming language which has been given

below. This function basically takes the Inverse Joukowski Transformation for each node, updates ζ_y, and then reapplies the Joukowski Transformation. This is done while accounting for the rotation and vertical motion of axes due to pitching and heaving.

```
#include "udf.h"
#include "dynamesh_tools.h"
#include "unsteady.h"
#include <math.h>
#define freq 0.5
#define angular_freq 2*M_PI*freq
#define alpha_max 10*M_PI/180
#define h_max 0.1
#define CofG_x -0.0529

DEFINE_GRID_MOTION(flapping_with_morphing,domain,dt,time,dtime)
{
Thread *tf = DT_THREAD(dt);
face_t f;
Node *v;
int n;

real h, alpha, h_current, alpha_current;
h = h_max*cos(angular_freq*time);
h_current = h_max*cos(angular_freq*(time-dtime));
alpha = alpha_max*sin(angular_freq*time);
alpha_current = alpha_max*sin(angular_freq*(time-dtime));

real R = 0.06;
real a = 0.05;
real zeta_x = -0.00625;
real zeta_y_0 = -0.01;
real zeta_y;
real zeta_y_current;
zeta_y = zeta_y_0*sin(angular_freq*time);
zeta_y_current = zeta_y_0*sin(angular_freq*(time-dtime));

SET_DEFORMING_THREAD_FLAG (THREAD_T0(tf));

begin_f_loop(f,tf)
{
    f_node_loop(f,tf,n)
    {
        v = F_NODE(f,tf,n);
        if (NODE_POS_NEED_UPDATE(v)) {
            NODE_POS_UPDATED(v);
```

```
real x, y, x_rel, y_rel, X, Y, X_1, Y_1, X_2, Y_2,
dist1, dist2, dist3, dist4, dist, x_new, y_new;
x = NODE_X(v);
y = NODE_Y(v);
x_rel = (x-CofG_x)*cos(alpha_current)
+ (y-h_current)*sin(alpha_current) + CofG_x;
y_rel = (y-h_current)*cos(alpha_current)
- (x-CofG_x)*sin(alpha_current);
//Accounting for rotation and vertical motion of axes

X_1 = 0.5*x_rel + 0.5*sqrt(0.5*(pow(x_rel,2)-pow(y_rel,2)
-4*pow(a,2)+ sqrt(pow((pow(x_rel,2)-pow(y_rel,2)-4*pow(a,2)), 2)
+ 4*pow(x_rel,2)*pow(y_rel,2))));
Y_1 = 0.5*y_rel + 0.5*sqrt(0.5*(-(pow(x_rel,2)-pow(y_rel,2)-
4*pow(a,2)) + sqrt(pow((pow(x_rel,2)-pow(y_rel,2)-4*pow(a,2)), 2)
+ 4*pow(x_rel,2)*pow(y_rel,2))));
X_2 = 0.5*x_rel - 0.5*sqrt(0.5*(pow(x_rel,2)-pow(y_rel,2)-
4*pow(a,2)+ sqrt(pow((pow(x_rel,2)-pow(y_rel,2)-4*pow(a,2)), 2)
+ 4*pow(x_rel,2)*pow(y_rel,2))));
Y_2 = 0.5*y_rel - 0.5*sqrt(0.5*( - (pow(x_rel,2)-pow(y_rel,2)
-4*pow(a,2)) + sqrt(pow((pow(x_rel,2)-pow(y_rel,2)-4*pow(a,2)), 2)
+ 4*pow(x_rel,2)*pow(y_rel,2))));;
//Inverse Joukowski Transformation

dist1 = fabs(pow(R,2) - (pow(X_1 - zeta_x, 2)
+ pow(Y_1 - zeta_y_current, 2)));
dist2 = fabs(pow(R,2) - (pow(X_2 - zeta_x, 2)
+ pow(Y_2 - zeta_y_current, 2)));
dist3 = fabs(pow(R,2) - (pow(X_1 - zeta_x, 2)
+ pow(Y_2 - zeta_y_current, 2)));
dist4 = fabs(pow(R,2) - (pow(X_2 - zeta_x, 2)
+ pow(Y_1 - zeta_y_current, 2)));

dist = dist1;
if (dist2 < dist) dist = dist2;
if (dist3 < dist) dist = dist3;
if (dist4 < dist) dist = dist4;

/*Choosing the appropriate branch of the
Inverse Joukowski Transformation*/

if (dist == dist1) {
    X = X_1;
    Y = Y_1;
}
else if (dist == dist2) {
```

```
                X = X_2;
                Y = Y_2;
            }
            else if (dist == dist3) {
                X = X_1;
                Y = Y_2;
            }
            else if (dist == dist4) {
                X = X_2;
                Y = Y_1;
            }

            Y = Y + zeta_y - zeta_y_current;
            //Update zeta_y

            x_rel = X + (pow(a,2)*X/(pow(X,2)+pow(Y,2)));
            y_rel = Y - (pow(a,2)*Y/(pow(X,2)+pow(Y,2)));
            //Reapplying the Joukowski Transformation

            x_new = (x_rel-CofG_x)*cos(alpha) - y_rel*sin(alpha) +  CofG_x;
            y_new = y_rel*cos(alpha) + (x_rel-CofG_x)*sin(alpha) + h;
            //Accounting for rotation and vertical motion of axes

            NODE_X(v) = x_new;
            NODE_Y(v) = y_new;
        }
    }
}
end_f_loop(f,tf);
}
```

This function was integrated in ANSYS Fluent using the Dynamic Meshing feature as mentioned above. Both smoothing and remeshing options were used.

For the setup, the inlet velocity is taken to be zero as discussed earlier and the outlet gauge pressure is also taken as zero. Transient option was used in the general settings and the SIMPLE pressure-velocity coupling scheme was used. The time step size was taken as 0.01 seconds and a total of 3000 time steps with a maximum of 20 iterations per step were run. At each time step the lift force and drag force were recorded, and after every 10 time steps the velocity and pressure contours were captured to create animations.

5 Results

The lift and drag force obtained from ANSYS Fluent simulation initially displays transientness but then settle down to a quasi-steady periodic pattern. Thus, we have only used the last 10 seconds of simulation results for analysis out of the

entire 30 seconds of simulation period, that is, after the quasi-steady state is reached. The plots for both lift and drag along with their comparison with the theoretical values have been shown below. As we can see, the time averaged drag is negative, that is, average thrust is being generated. Time average lift, however, is zero due to symmetry in the motion of the airfoil.

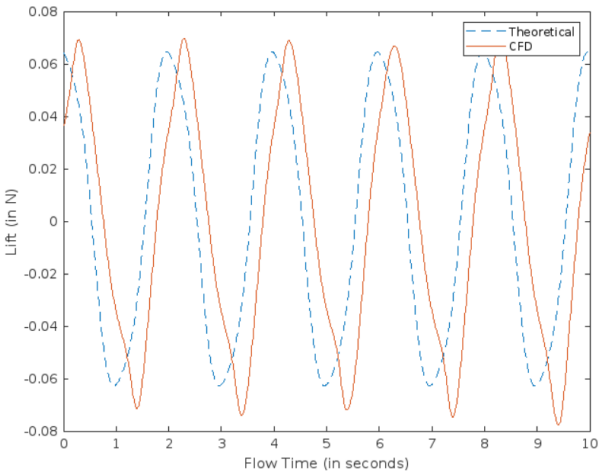


Figure 7: Comparison of Lift Force obtained from theory and simulation

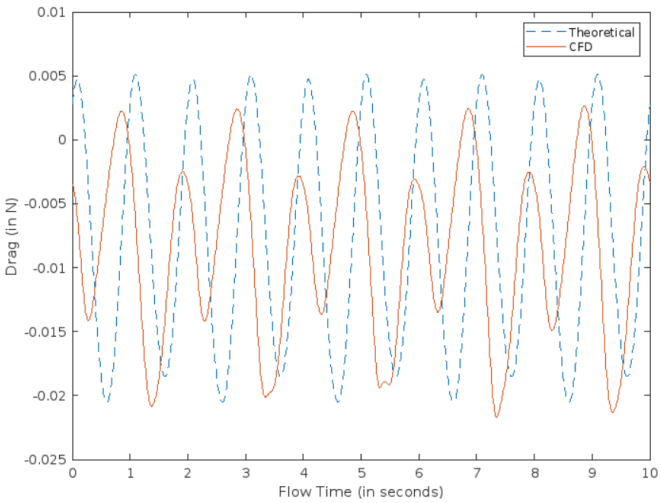


Figure 8: Comparison of Drag Force obtained from theory and simulation

The pressure and velocity contour animations can be downloaded from the links below.

Velocity Contour Animation: `https://github.com/ishaansharma2903/lsa-term-paper/blob/733d3f73d08b376199178a0b42c20967e3d5b923/velocity_contours.mp4`

Pressure Contour Animation: `https://github.com/ishaansharma2903/lsa-term-paper/blob/733d3f73d08b376199178a0b42c20967e3d5b923/pressure_contours.mp4`

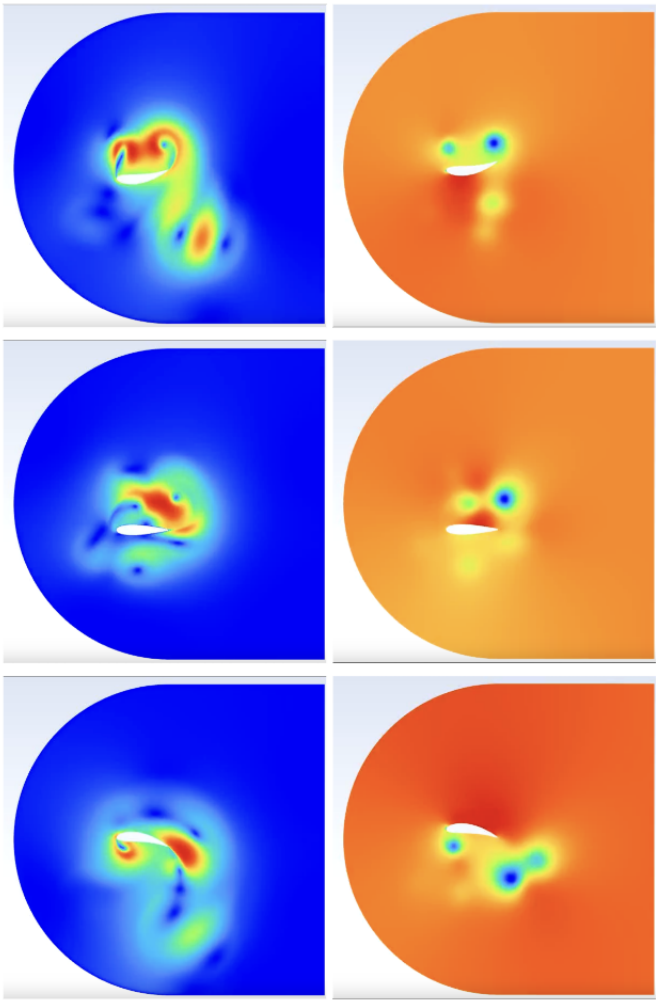


Figure 9: Velocity and Pressure contours at different time instants

6 Conclusion

In this paper, we studied the thrust and lift generated by a periodically heaving, pitching and morphing Joukowski Airfoil. We first derived the analytical model using potential flow theory. Finally, we compared the thrust and lift generated in the analytical model with ANSYS Fluent Simulations. For the motion functions selected ($h = h_0 \cos(\omega t)$, $\alpha = \alpha_0 \sin(\omega t)$ and $\zeta_y = -\zeta_{y_0} \sin(\omega t)$ the analytical model predicted a negative time-averaged drag (or positive time-averaged thrust) and zero time-averaged lift which agrees with the results of the simulation. The peak magnitudes obtained by the analytical model are also fairly close to those obtained from simulations.

References

[1] Joel Guerrero. "Numerical simulation of the unsteady aerodynamics of flapping flight". PhD thesis. 2009. URL: https://hdl.handle.net/11567/1020121.

[2] Shaobin Li, Zhenxin Tao, and Xizhen Song. "Unsteady Lift Model for Morphing Airfoil Based on Potential Flow Theory". In: *Journal of Aerospace Engineering* 31.2 (2018), p. 04018006. DOI: 10.1061/(ASCE)AS.1943-5525.0000820. eprint: https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29AS.1943-5525.0000820. URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29AS.1943-5525.0000820.

[3] Richard James Mason. "Fluid Locomotion and Trajectory Planning for Shape-Changing Robots." PhD thesis. California Institute of Technology, 2003. DOI: doi:10.7907/MFM1-0866. URL: https://resolver.caltech.edu/CaltechETD:etd-05292003-160843.

[4] Knut Streitlien. *A simulation procedure for vortex flow over an oscillating wing*. Tech. rep. MIT Sea Grant College Program report ; MITSG 94-7. Massachusetts Institute of Technology, Sea Grant College Program,;National Sea Grant College Program (U.S.);United States, Advanced Research Projects Agency, 1994. URL: https://repository.library.noaa.gov/view/noaa/9917.

[5] Adam Ysasi, Eva Kanso, and Paul K. Newton. "Wake structure of a deformable Joukowski airfoil". In: *Physica D: Nonlinear Phenomena* 240.20 (2011). Special Issue: Fluid Dynamics: From Theory to Experiment, pp. 1574–1582. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2011.06.021. URL: https://www.sciencedirect.com/science/article/pii/S0167278911001783.