Lift and Drag coefficient in a section of a plane's wing

Xavier Cabanes and Alex Nowak
(Dated: May 24, 2016)

In this article we face a very challenging numerical problem. We aim to calculate the lift and drag coefficient for a symmetrical airfoil and try to model, as much as our computational power allows us, both coefficients for the specific case of a plane's takeoff. We see how difficult this problem is due to the turbulences and sharp fronts on the boundary layer appearing for high Reynolds numbers which are impossible to capture without a good numerical implementation and a huge computational power.

I. NAVIER-STOKES EQUATIONS

Fluid dynamics are governed by Navier-Stokes equations. These equations are very famous both for their importance and complexity. Clay Math Institute states that the proof for the existence and smoothness of the solution of NS in $\mathbb{R}^3$ is one of the seven most important mathematical problems of the millennium. Understanding the behaviour of these solutions is essential for many fluid applications.

Derivation of NS follows from the conservation of momentum

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + s = 0 \tag{1}$$

where $\mathbf{u}$ is the fluid velocity, $\rho\mathbf{u}$ the mass flux and s corresponds to momentum sources. Rearranging and using the necessary operators identities leads to

$$\mathbf{u}\left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u})\right) + \rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = s \tag{2}$$

The first term vanishes applying mass conservation and the second term multiplying the fluid density is called the material derivative $\frac{D\mathbf{u}}{Dt}$ and we can see

$$\rho \frac{D\mathbf{u}}{Dt} = s \tag{3}$$

as an extension of Newton's second law ($\mathbf{F} = m\mathbf{a}$) for continuum media. We can now split the momentum source s into one corresponding to surface forces $\nabla \cdot \boldsymbol{\sigma}$ and another for body forces such as gravity $\mathbf{f}$, i.e, $s = \nabla \cdot \boldsymbol{\sigma} + f$. Surface forces or stresses can be taken into account using the Cauchy stress tensor $\boldsymbol{\sigma}$, which is a second order symmetric tensor. $\boldsymbol{\sigma}$ sends a unitary direction $\mathbf{n}$ in space to the vector $\mathbf{T}^{(n)}$ corresponding to the stress across the plane with normal vector $\mathbf{n}$.

The stress tensor can be split up into two terms ($\boldsymbol{\sigma} = \boldsymbol{\tau} - p$) by separating the contribution of the mechanical pressure of the fluid $p = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz})$, defined by minus the mean normal of the stress, and the shear stress tensor $\boldsymbol{\tau}$ which describes only shear stresses of the fluid.

We finally get to the most general form of fluid dynamics

$$\rho \frac{D\mathbf{u}}{Dt} + \nabla p = \nabla \cdot \boldsymbol{\tau} + \mathbf{f} \tag{4}$$

However, this equation is not ready for use due to the shear stresses $\boldsymbol{\tau}$ which are unknown, i.e, we need a constitutive equation by restricting us to specific fluid families. We will focus on incompressible isotropic Newtonian fluids. Incompressibility assumes

$$\nabla \cdot \mathbf{u} = 0 \tag{5}$$

Incompressible newtonian fluids are those whose shear stresses are of the form

$$\boldsymbol{\tau} = 2\mu\boldsymbol{\epsilon} = 2\mu\nabla^S\mathbf{u} = \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T) \tag{6}$$

i.e, newtonian fluids are fluids in which the viscous stresses arising from its flow are proportional to the local strain rate (rate of change of deformation over time). This constant of proportionality μ is called dynamic viscosity of the fluid $[Pa \cdot s]$.

In our project we will be solving stationary Navier-Stokes equation, so $\partial u/\partial t = 0$. With all these assumptions, we finally get to the famous stationary Navier-Stokes equations for incompressible fluids,

Problem 1. (Classical steady N-S)
Find $\boldsymbol{u} \in [C^2(\Omega)]^2$ *and* $p \in \boldsymbol{C}^1(\Omega)$ *such that*

$$\begin{cases} (\mathbf{u} \cdot \nabla)\,\mathbf{u} + \nabla p - \frac{1}{Re}\nabla^2\mathbf{u} = f \text{ in } \Omega \\ \nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \\ \mathbf{u} = \mathbf{u}_D \text{ on } \Gamma_D \\ \mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{t} \text{ on } \Gamma_N \end{cases}$$

with $\Gamma_D \cup \Gamma_N = \partial\Omega$ *and* $\sigma = (\frac{1}{Re}\nabla\boldsymbol{u} - p)$

In Problem 1, the equations have been non-dimensionalized and all the parameters reduced to the Reynolds number

$$Re = \frac{\rho v L}{\mu} \tag{7}$$

with L being a characteristic length of the fluid and v is the maximum velocity of the object relative to the fluid. The Reynolds number helps to predict different flow patterns in different fluid situations, for instance, higher Re implies dominant convection term, hence more turbulent fluid.

Remark: for low Reynolds numbers, the Navier-Stokes equation can be simplified to the Stokes equation, which does not take into account the contribution of the convective term $(\mathbf{u}\cdot\nabla)\mathbf{u}$ and so it becomes a diffusion equation. This simplification is usually used as a previous "training" case of the Navier-Stokes equation since it becomes a linear equation, and so we have done in this project.

II. IMPLEMENTATION

Since it was our first attempt to solve fluid dynamics numerical problems, our first step in this project was to develop a code to solve the Stokes problem. Then we solved the Navier-Stokes equation and in the end we implemented the routine to integrate the Lift and Drag coefficient in our wing profile. So, we will first be describing the FEM method and the problem to be solved.

Finite Element Method (FEM) is the most used and popular method for numerically solving PDEs. The power of FEM comes from its easy implementation but overall for its applicability to a general domain Ω. FEM method consists in solving the weak formulation of Problem 1, i.e, reformulating the problem in an integral form by taking the scalar product $(\cdot,\cdot)$

$$(u,v)=\int_\Omega uv d\Omega$$

in $\mathcal{L}_2(\Omega)$ at the equation and forcing the identity for all the functions of a properly chosen space. The weak form of Problem 1 reads,

Problem 2. (Weak form of N-S)
Find $\boldsymbol{u}\in\boldsymbol{\mathcal{V}}_D$ and $p\in\mathcal{Q}$ such that for all $\boldsymbol{w}\in\boldsymbol{\mathcal{V}}$ and all $q\in\mathcal{Q}$

$$\begin{cases} a(\mathbf{w},\mathbf{u})+c(\mathbf{w},\mathbf{u},\mathbf{u})+b(\mathbf{w},p)=(\mathbf{w},\mathbf{f})+(\mathbf{w},\mathbf{t})_{\Gamma_N} \\ b(\mathbf{u},q)=0 \end{cases}$$

with $a(\cdot,\cdot)$, $c(\cdot,\cdot,\cdot)$ and $b(\cdot,\cdot)$ being linear functionals on its arguments defined by $a(\mathbf{w},\mathbf{u})=\frac{1}{Re}\int_\Omega(\nabla\mathbf{w}:\nabla\mathbf{u})d\Omega$, $b(\mathbf{v},q)=-(q,\nabla\cdot v)$ and $c(\mathbf{w},\mathbf{u},\mathbf{u})=(\mathbf{w},(\mathbf{u}\cdot\nabla)\mathbf{u})$, and $\boldsymbol{\mathcal{V}}_D=\{\mathbf{w}\in\boldsymbol{\mathcal{V}}\text{ s.t }\mathbf{w}_{|\Gamma_D}=\mathbf{u}_D\}$.

In the case of the FEM implementation, the functional spaces are $\boldsymbol{\mathcal{V}}=(\mathcal{S}_k^h)^2$ and $\boldsymbol{\mathcal{V}}_D=[\mathcal{S}_k^h]_D^2$ for the velocity and $\mathcal{Q}=\mathcal{S}_{k-1}^h$ for the pressure, where $\mathcal{S}_k^h=\text{span}_{i\in[N_u]}\{\psi_i^k\}$. These functions ψ_i^k are called shape functions, and are piecewise continuous degree-k polynomials indexed by $[N_u]=\{1,\dots,N_u\}$ (indexes corresponding to set of nodes) that have the property of being Kronecker's delta on the nodes, so $\psi_i^k(x_j)=\delta_{ij}\ \forall i,j\in[N_u]$.

Then we can express our approximate FEM solution for the velocity and pressure as elements of these discretized spaces in function of their nodal values $\bar{\boldsymbol{u}}_{\boldsymbol{i}}\in\mathbb{R}^2$ and p_i, $\mathbf{u}\approx\mathbf{u}^h=\sum_{i=1}^{N_u}\bar{\boldsymbol{u}}_{\boldsymbol{i}}\psi_i^k$ and $p\approx p^h=\sum_{i=1}^{N_p}p_i\psi_i^{k-1}$.

Once all the integrals are computed, we get an equation of the form $\mathbf{F}(\mathbf{U},\mathbf{P})=0$, with $\mathbf{U}$ and $\mathbf{P}$ being vectors containing all the nodal values. To find the solution of such equation, for instance, a Newton-Raphson method with analytical gradients can be used since it is usually a non-linear equation.

Remark: The true power of FEM method comes from the technique it uses to compute the integrals. When integrating the product of two shape functions ψ_i and ψ_j we only need to integrate over few elements rather than in the whole mesh because of its compact support. This property is exploited by computing the integrals separately for every element and then adding the value of the local integral to the main integral in which it contributes, calling this process *assembly.*
The integral over every element will be computed using a simple change of variables called the *isoparametric transformation* which is very easy to implement and it transforms a specific element in the mesh into a *reference element* for which we already know its gaussian points and weights and the numerical integral becomes straightforward. For further information, see [1].

A. The Stokes problem

As we can see in the remark of Problem 1, making the convective term equal zero transforms the equation into a linear one.
Finally, the problem to be solved is the same as Problem 2 but without the linear functional $c(\mathbf{w},\mathbf{u},\mathbf{u})$. After writing the nodal equations for each of the shape functions, the zero of the following function must be found:

$$F(\mathbf{U},\mathbf{P})=\left(\begin{array}{c|c} K & G \\ \hline G^T & 0\end{array}\right)\cdot\begin{pmatrix}\mathbf{U}\\ \mathbf{P}\end{pmatrix}-\begin{pmatrix}\mathbf{f}\\ \mathbf{h}\end{pmatrix} \tag{8}$$

Solving this system brings to the solution $\mathbf{U}=K^{-1}(\mathbf{f}-G\mathbf{P})$ and $(G^TK^{-1}G)\mathbf{P}=G^TK^{-1}\mathbf{f}-\mathbf{h}$ which, as told before, corresponds to solve lynear systems of equations.

We are not going to go deep on the details of how to obtain these matrices and vectors, but in the following table we can see which is the relation between them and the different operators present in the problem formulation:

Matrix or vector	Operator
K	$a(\mathbf{w},\mathbf{u})$
G	$b(\mathbf{w},\mathbf{p})$
G^T	$b(\mathbf{u},\mathbf{q})$
$\mathbf{f}$	$(\mathbf{w},\mathbf{q})-a(\mathbf{w},\mathbf{v_D})+(\mathbf{w},\mathbf{t})_{\Gamma_N}$
$\mathbf{h}$	$-b(\mathbf{v_D},\mathbf{q})$

B. The Navier-Stokes problem

Now the problem to be solved is Problem 3. In this case, the function to which we must find a zero is just the same as the Stokes problem but adding the convective term. This translates into finding the zero of the the function of the velocity field and the pressure:

$$F(\mathbf{U},\mathbf{P}) = \left(\begin{array}{c|c} K + C(\mathbf{U}) & G \\ \hline G^T & 0 \end{array}\right) \cdot \begin{pmatrix} \mathbf{U} \\ \mathbf{P} \end{pmatrix} - \begin{pmatrix} \mathbf{f} \\ \mathbf{h} \end{pmatrix} \quad (9)$$

As we can see, in this case there is no way we can compute the solution to this zero-finding problem just by solving a system of linear equations because of the term $C(\mathbf{U})$. In this case, a Newton-Rhapson method is used in order to find the zero of this equation. Further details can be found at [2].

III. LIFT AND DRAG COEFFICIENTS

As told before, our main goal in this article is to compute two important aerodynamic coefficients; C_L (Lift coefficient) and C_D (Drag coefficient). Let us introduce these concepts:

Definition 1. We define the total traction force T of the fluid on a volume Ω as the integral of the traction $\mathbf{t} = \mathbf{n} \cdot \boldsymbol{\sigma}$ on the volume surface

$$T = \int_{\partial\Omega} \mathbf{t} dS = \int_{\partial\Omega} \mathbf{n} \cdot \boldsymbol{\sigma} dS \quad (10)$$

We then directly define the drag coefficient as the horizontal component of the total traction $C_D = T \cdot e_1$, i.e, the force that the fluid does against the horizontal motion of the volume.

Analogously, we define the lift force as the vertical component $C_L = T \cdot e_2$, i.e, the total vertical force that the fluid exerts to the volume. where $\mathbf{n}$ is the normal vector pointing outwards the volume. We can see it graphically in the following figure:

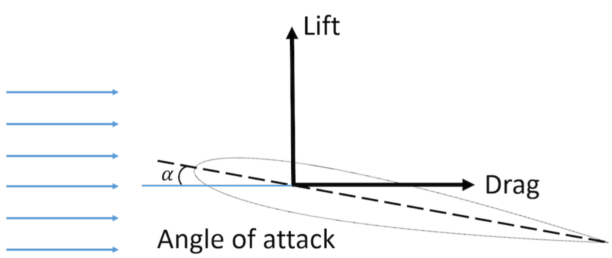


FIG. 1: Representation of the Lift and Drag force, and the angle of attack α.

T depends on σ, and therefore it depends on $\mathbf{u}$ and p ($\boldsymbol{\sigma} = \nu\nabla\mathbf{u} - p$), so we must first calculate the solution of Problem 3, and then integrate on the boundary of the airfoil.

In the implementation, the elements to be taken into account are not triangle-shaped (2D) but pieces of curves on the airfoil (1D). Then, the integral is calculated taking the isoparametric transformation into account and using a suitable quadrature over the points of each element.

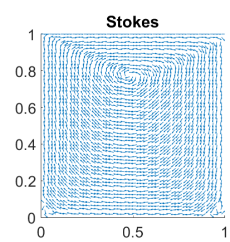
IV. RESULTS

A. Own code result: cavity flow

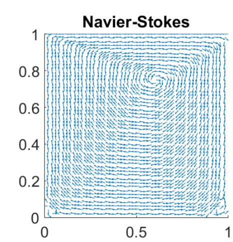
Before facing the modeling of the air flow around the NACA012 airfoil profile, we first solved NS inside a squared cavity, i.e, $\Omega = [0,1]^2$. We put null Dirichlet conditions at all faces of Ω unless the top one, in which we fixed a fixed positive velocity.

We used this problem to compare between Stokes and Navier-Stokes solution and to observe the main difference in the flow behavior in both cases.

The Reynolds number used in the Navier-Stokes solution is the biggest we could use in order to have a reasonable computational time in our `Matlab` code. We observe that the central swirl is moved rightwards when added the convective term.



(a) Normalized velocities of Stokes cavity solution.



(b) Normalized velocities of Navier-Stokes cavity solution with Re = 80.

B. NACA012 airfoil profile

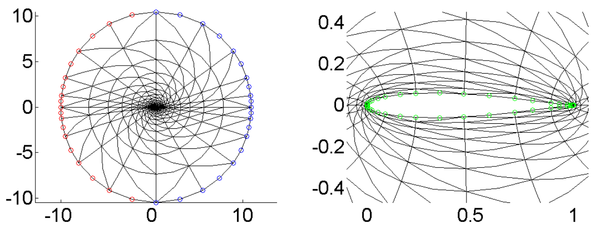
Remark: After developing all these codes by ourselves, we encountered problems dealing with CPU time to get proper results for realistic values of the Reynolds number, mesh sizes and angle of attack.

That's why we used an already developed code in `Fortran` and, although these results do not come from our own code, all the previous work has been done and conveniently checked and without it the didactic objective of this project would not be achieved.

We chose our domain Ω to be a circle limited by NACA012 profile [3] in the origin. The boundary in our case is the circumference (Γ_c) and the wing with NACA012 profile, (Γ_w). The left and right part of the circumference are called inlet ($\Gamma_{c,in}$) and outlet ($\Gamma_{c,out}$) respectively, and have different boundary conditions each. These conditions are: $\mathbf{u} = u_0(\cos\alpha, \sin\alpha)$ on $\Gamma_{c,in}$, $\mathbf{n} \cdot \boldsymbol{\sigma} = 0$ on $\Gamma_{c,out}$, $\mathbf{u} = 0$ on Γ_w.

The zero velocity condition taken in the wing profile is called non-slip condition. Applying this condition creates a layer near the boundary called *boundary layer* where the magnitude of the velocity increases dramatically. This increase is difficult to be captured without having the boundary properly discretized. We will see how this is solved in the next figures.

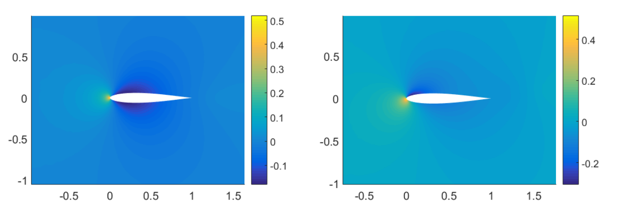
Another thing to be taken into account is the *mesh*. As told in a previous section, the domain is cut in triangular "pieces" called elements, see Figure 2.



(c) Whole mesh (d) Near wing mesh

FIG. 2: FE mesh for the solution of the dimensionless problem and boundary nodes corresponding to the *inlet* (red), *outlet* (blue) and *wing* (green).

Note that the size of the element decreases as we get closer to the boundary layer. This is done to properly capture the spike in the velocity field magnitude near the boundary on the airfoil.
As an example Figure 3 shows the pressure for two different values of α.



(a) $\alpha = 0^o$ (b) $\alpha = 5^o$

FIG. 3: Pressure field for different values of α.

Higher pressure near the front of the profile can be appreciated. For case $\alpha = 5^o$, the pressure under the wing is greater than over the wing, thus the Lift force will be *upwards*.

As the main objective of this project was computing the Lift and Drag coefficients, we ran the code for different angles of attack $\alpha \in [-5, 5]$. In Figure 4 we can see that the Lift force increases as the angle of attack increases. This is as we would expect, since during take-off the angle of attack is greater than during the flight because the Lift force must be enough to counter gravity and also make the airplane fly.

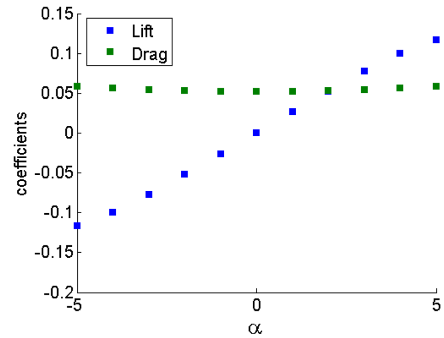


FIG. 4: Lift and drag coefficients for the solution of Problem 2 with $Re = 5000$.

Let us make an example we could have with these conditions: $\rho_0 = 1.2041\ kg/m^3$, $\mu = 18.27 \cdot 10^{-6}\ Pa \cdot s$, $L = 0.04\ m$ and $V = \frac{Re\mu}{\rho L} = 1.896645\ m/s$, $\alpha = 5^o$ and $C_l = 0.1168236$. Assuming we have an airplane with wings of length L each and speed V, we get a final lift force: $[C_l] = \left[\frac{F}{L}\right] = \left[\frac{L^4\rho}{LT^2}\right] = [L\rho V^2] = [\mu Re V]$, so $C_{l,dim} = C_{l,ndim} \cdot \mu Re V = 0.02024067\ N/m$. So our final force becomes $F_l = 2C_{l,dim}L = 0.009715524\ N$ which is equivalent to the weight of a $97.15g$ airplane. Assuming it is made of carbon fiber ($\rho_{cf} \approx 1.55\ g/cm^3$) we could have a volume of $62.68cm^3$.

V. CONCLUSION

In this project we introduce the reader to the FEM method for modeling fluid dynamics and its main difficulties. We find that the Lift coefficient grows linearly with respect to the angle of attack of the airfoil.

We have shown the overwhelming computational power that it is needed in order to model fluids flow at high velocity in situations like a plane's take-off. In the latter case, Reynolds number is of the order of 10^6, three orders of magnitude larger than the one we computed using an efficient implementation in `Fortran` and four orders of magnitude larger than the one we used for the cavity flow in our `Matlab` code.

We also found a toy example for which we could use our 5000 Reynolds computation and observe how far still we are to a real plane's take-off modeling. However, the correct model for the whole airplane must take into account the flow dependence through different wing sections, leading to a 3-D mesh modelling and huge increase of computational power which can only be handled with the help of supercomputers.

[1] *Numerical Mathematics* Quarteroni, A; Sacco, R; Saleri, Fausto (2000)
[2] *Finite Element Methods for Flow Problems* Donea,J; Huerta, A (2003)
[3] http://airfoiltools.com/airfoil/details?airfoil=n0012-il