# AEM4253/5253–Computational Fluid Mechanics

Fall Semester 2022

*Final Project*

*The development of a* ***Compressible Euler Equation*** *solver.*

Suryanarayan(Surya) Ramachandran
email:ramac106@umn.edu

July 23, 2023

*"As soon as we started programming, we found to our surprise that it wasn't as easy to get programs right as we had thought. Debugging had to be discovered. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs." – Maurice Wilkes (British Computer Scientist who helped develop one of the earliest stored program computers) discovers debugging, 1949*

## 1 Domain, Geometry and Mesh

We operate in elliptic co-ordinates for this problem for which the transformation from the computational space to the physical space is given by:

$$x = \cosh\eta\cos\xi$$

$$y = \sinh\eta\sin\xi$$

We note that eliminating $\xi$ fetches us the equation of an ellipse as follows:

$$\left(\frac{x}{\cosh\eta}\right)^2 + \left(\frac{y}{\sinh\eta}\right)^2 = \sin^2\xi + \cos^2\xi = 1$$

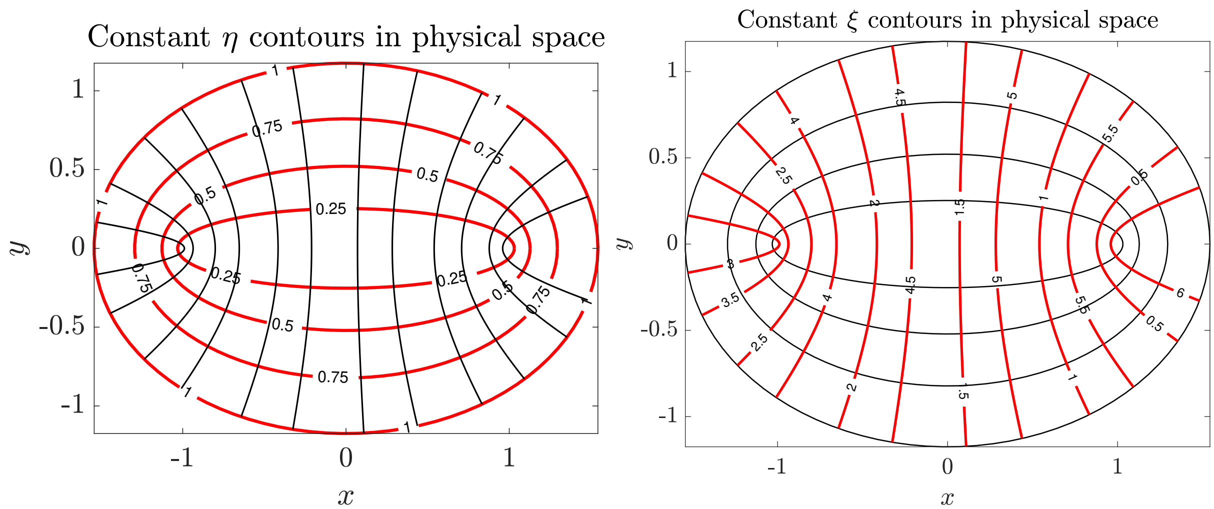
Which resembles:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

Hence it is easy to see here that the $\xi$-parameter basically controls the angular location, while the $\eta$ parameter, in some sense, controls the radial location. From this understanding that we have gathered, we hence note that $\xi \in [0, 2\pi]$. The constant $\eta$ contours are ellipses and the constant $\xi$ contours can proven to be hyperbolas:

$$\left(\frac{x}{\cos\xi}\right)^2 - \left(\frac{y}{\sin\xi}\right)^2 = \cosh^2\eta - \sinh^2\eta = 1$$

For an arbitrary choice of $\eta \in [-1, 1]$, we try to visualize the mesh in the physical space using MATLAB. The constant $\eta$ and $\xi$ contours are visualized below in Fig. 1: We note that as $\eta$ increases the ellipse gets closer to a circle and hence the ratio of the major to the minor axis




(a) (b)

**Figure 1:** *Constant $\eta$ and $\xi$ contours visualized in red in the physical space*

decreases. Hence to cover only the exterior portion of a 4-to-1 ellipse, we impose the condition that:

$$a \leq 4b$$

Based on the reference ellipse equation defined earlier. Hence, for our system of co-ordinates adopted:

$$\cosh \eta \leq 4 \sinh \eta \implies \tanh \eta \geq 0.25$$

Solving this inequality fethes us limits of $\eta$ as:

$$\eta \in [\tanh^{-1}(0.25), \infty)$$

At the surface of the elliptical body, we note that:

$$\eta = \tanh^{-1}(0.25)$$

Alternatively, this can be done by just excluding the points inside the bounding ellipse by inverting the condition above. If $D$ is the set of all points outside the ellipse, we see that D can be given by:

$$D = \{(\xi, \eta) : \tanh \eta < 0.25\}$$

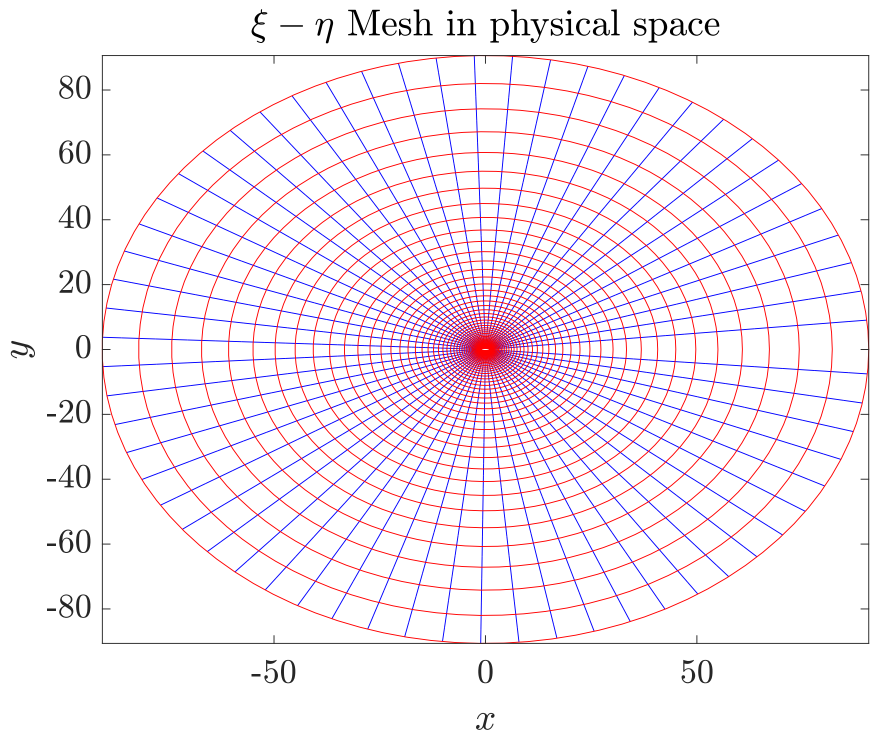
Since we are solving the problem computationally, we also have to decide on the upper limit for $\eta$. To do so, we note that at $\eta = \tanh^{-1}(0.25)$, the length of the semi-major axis of the ellipse is $\cosh(\tanh^{-1}(0.25)) \approx 1.0328$. Following the usually recommended CFD practice guidelines, a safe value of the length of the domain would be 6 times the length of the body being simulated [1]. Thus, we consider an upper bound of $\approx 5 \times 1.0328 \approx 5.2$. Hence the final limits for $(\xi, \eta)$ are:

$$\xi \in [0, 2\pi]$$

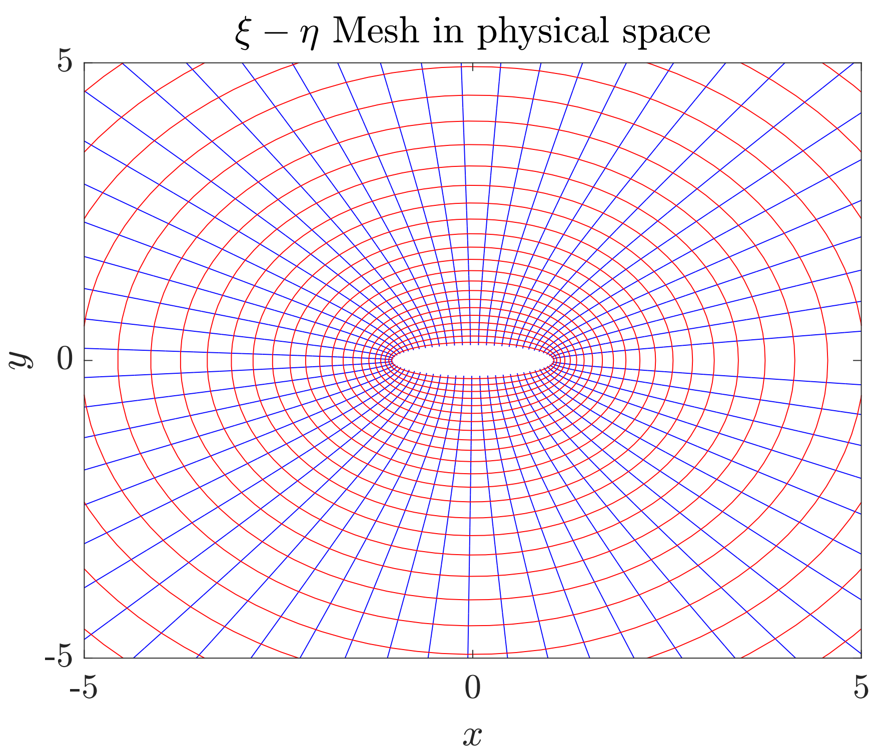
$$\eta \in [\tanh^{-1}(0.25), 6.2]$$

The mesh presented Fig 1 is regenerated with 100 points along both $\xi$ and $\eta$ between the limits computed above in Figs 2 and we see that the limits work well to capture the domain outside the ellipse with mentioned aspect ratio of 4. We hence note that the resolution $(\Delta\xi, \Delta\eta)$=$(0.635, 0.051)$.

We can now compute the mesh metrics for obtaining the Jacobian. This is detailed in the next section.




(a)




(b)

**Figure 2:** *Final Mesh and Domain. The red and blue lines represent lines of constant $\eta$ and $\xi$ respectively.*

To double-check if we are getting the aspect ratio right, we zoom further into the domain and get the major and minor axes from the plot below and we roughly see that this is indeed 4. We

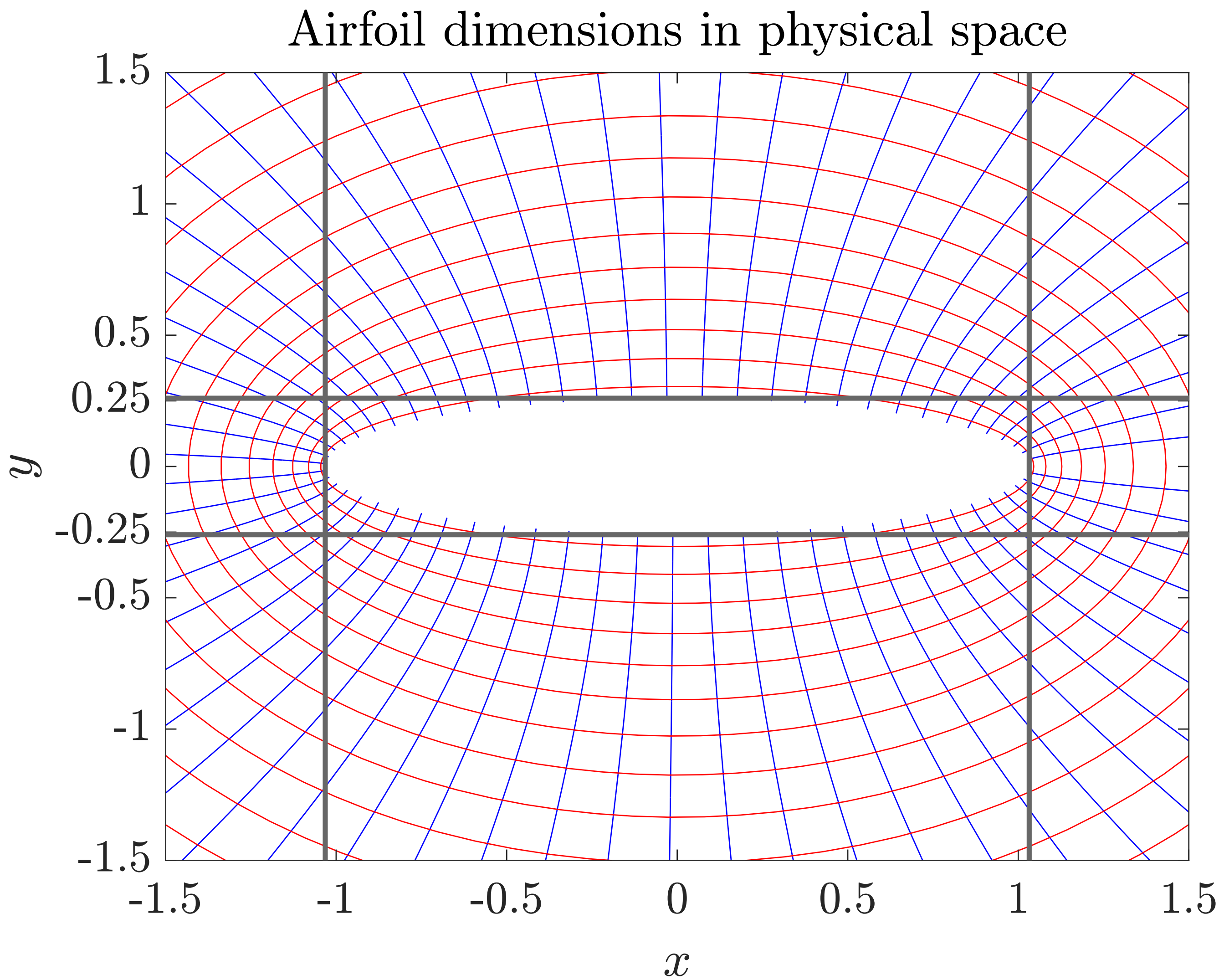



**Figure 3:** *We note that the constant-$\xi$ lines terminate at approximately the intended aspect ratio of the airfoil (4).*

set the MATLAB arrays `xi(N,N)` and `eta(N,N)` such that the rows of `xi(N,N)` traverse along an ellipse and the columns of `xi(N,N)` jump between ellipses in the outward direction. Hence `xi(i,j)` indexing follows index $i$ for looping along the ellipse (in the anticlockwise sense) and index $j$ for choosing which ellipse to traverse through. Hence every column of `xi(N,N)` is a new ellipse going away from the airfoil.

## 2 Transformation Matrices/Jacobian

To transform the derivatives from $(x, y)$ to $(\xi, \eta)$ we use the chain rule ( Here $f_\xi = \frac{\partial f}{\partial \xi}$):

$$\begin{pmatrix} f_x \\ f_y \end{pmatrix} = \begin{pmatrix} \xi_x & \eta_x \\ \xi_y & \eta_y \end{pmatrix} \begin{pmatrix} f_\xi \\ f_\eta \end{pmatrix} = J \begin{pmatrix} f_\xi \\ f_\eta \end{pmatrix}$$

Similarly:

$$\begin{pmatrix} f_\xi \\ f_\eta \end{pmatrix} = \begin{pmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{pmatrix} \begin{pmatrix} f_x \\ f_y \end{pmatrix} = J^{-1} \begin{pmatrix} f_x \\ f_y \end{pmatrix}$$

We compute $J^{-1}$ first and then invert it to get back $J$ using:

$$J = \begin{pmatrix} \xi_x & \eta_x \\ \xi_y & \eta_y \end{pmatrix} = (J^{-1})^{-1} = \begin{pmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{pmatrix}^{-1} = \frac{1}{(x_\xi y_\eta - x_\eta y_\xi)} \begin{pmatrix} y_\eta & -y_\xi \\ -x_\eta & x_\xi \end{pmatrix}$$

To obtain these gradient terms of the matrix, we use a central-difference scheme for the interior points:

$$(x_\xi)_{i,j} = \frac{x_{i+1,j} - x_{i-1,j}}{2\Delta\xi}$$

and a first-order backward/forward scheme for the boundary points(at the airfoil and at the end of the domain respectively). Following MATLAB indexing (starts at 1):

$$(x_\xi)_{1,j} = \frac{x_{2,j} - x_{1,j}}{\Delta\xi}$$

if $N$ denotes the last index:

$$(x_\xi)_{N,j} = \frac{x_{N,j} - x_{N-1,j}}{\Delta\xi}$$

The same scheme/method is applied to compute other derivatives.

# 3 Solution to the Euler Equations

## 3.1 Transformed Equations

We solve for the $(\xi, \eta)-$space Euler Equations given by:

$$\frac{\partial \tilde{U}}{\partial t} + \frac{\partial \tilde{F}}{\partial \xi} + \frac{\partial \tilde{G}}{\partial \eta} = 0$$

Where the transformed variables are related to the original variables via the relation:

$$\tilde{U} = \frac{U}{J}, \tilde{F} = \frac{\xi_x F + \xi_y G}{J}, \tilde{G} = \frac{\eta_x F + \eta_y G}{J}$$

We exploit the homogeneity of the flux vectors $F, G$ of the Euler equations for which we wrote (in the $x, y-$space):

$$F(U) = \frac{\partial F}{\partial U} U = AU$$

$$G(U) = \frac{\partial G}{\partial U} U = BU$$

Where $A, B$ are the Jacobians for the fluxes. Writing the corresponding Jacobians for the transformed space:

$$\tilde{F}(\tilde{U}) = \frac{\partial \tilde{F}}{\partial \tilde{U}} \tilde{U} = \tilde{A}\tilde{U}$$

$$\tilde{G}(\tilde{U}) = \frac{\partial \tilde{G}}{\partial \tilde{U}}\tilde{U} = \tilde{B}\tilde{U}$$

Where $\tilde{A}$ for instance is given by:

$$\tilde{A} = \frac{\partial}{\partial \tilde{U}}\left(\frac{\xi_x F(U) + \xi_y G(U)}{J}\right)$$

Using the fact that: $\tilde{U} = U/J$ and hence $\frac{\partial}{\partial \tilde{U}} = J\frac{\partial}{\partial U}$:

$$\tilde{A} = J\frac{\partial}{\partial U}\left(\frac{\xi_x F(U) + \xi_y G(U)}{J}\right)$$

$$= \xi_x \frac{\partial F}{\partial U} + \xi_y \frac{\partial G}{\partial U}$$
$$= \xi_x A + \xi_y B$$

Similarly:

$$\tilde{B} = \eta_x A + \eta_y B$$

## 3.2 Flux Vector Splitting in Transformed Space

### 3.2.1 Methodology

We now need to split the fluxes $\tilde{F}$ and $\tilde{G}$ as :

$$\tilde{F} = \tilde{F}^+ + \tilde{F}^- = \tilde{A}^+\tilde{U} + \tilde{A}^-\tilde{U}$$

$$\tilde{G} = \tilde{G}^+ + \tilde{G}^- = \tilde{B}^+\tilde{U} + \tilde{B}^-\tilde{U}$$

Where $\tilde{A}^+$ and $\tilde{A}^-$ are obtained using an eigen decomposition:

$$\tilde{A} = \tilde{A}^+ + \tilde{A}^-$$

$$= \tilde{R}\tilde{\Lambda}^+\tilde{R}^{-1} + \tilde{R}\tilde{\Lambda}^-\tilde{R}^{-1}$$

Where:

$$\tilde{\Lambda} = \tilde{\Lambda}^+ + \tilde{\Lambda}^- = \underbrace{\left(\frac{\tilde{\Lambda} + |\tilde{\Lambda}|}{2}\right)}_{\tilde{\Lambda}^+} + \underbrace{\left(\frac{\tilde{\Lambda} - |\tilde{\Lambda}|}{2}\right)}_{\tilde{\Lambda}^-}$$

Due to problems of discontinuity with the absolute value function, we note that a gentler transition can be obtained using:

$$\tilde{\lambda}^{\pm} = \frac{\tilde{\lambda} \pm \sqrt{\tilde{\lambda}^2 + \epsilon^2}}{2}$$

Now, we know that $\tilde{A}$ satisfies:

$$\tilde{A} = \xi_x A + \xi_y B$$

$$= \left[\left(\frac{\xi_x}{\sqrt{\xi_x^2 + \xi_y^2}}\right) A + \left(\frac{\xi_y}{\sqrt{\xi_x^2 + \xi_y^2}}\right) B\right]\sqrt{\xi_x^2 + \xi_y^2}$$

$$= (n_x A + n_y B) r_\xi$$
$$= A' r_\xi$$

Where we now note that:

$$n_x^2 + n_y^2 = 1$$

The same arguments can be made for $\tilde{G}$ as well, except we have:

$$n_x = \frac{\eta_x}{\sqrt{\eta_x^2 + \eta_y^2}}$$

$$n_y = \frac{\eta_y}{\sqrt{\eta_x^2 + \eta_y^2}}$$

And hence $r_\xi$ becomes $r_\eta = \sqrt{\eta_x^2 + \eta_y^2}$. We can easily relate the eigen-decomposition of $A'$ to that of $\tilde{A}$ as the two matrices just differ by a scalar multiplier $r_\xi$. Using the definitions of the eigen-decomposition of $A'$ and $\tilde{A}$:

$$A' = R' \Lambda' R'^{-1}$$
$$\tilde{A} = \tilde{R} \tilde{\Lambda} \tilde{R}^{-1}$$

And the fact that:

$$\tilde{A} = r_\xi A' = r_\xi (R' \Lambda' R'^{-1}) = R' (\Lambda' r_\xi) R'^{-1} = \tilde{R} \tilde{\Lambda} \tilde{R}^{-1}$$

Thus the relation between the eigen-decompositions are obtained as:

$$\tilde{R} = R'$$

$$\tilde{R}^{-1} = R'^{-1}$$

$$\tilde{\Lambda} = r_\xi \Lambda'$$

In simpler words, the eigen-vectors remain the same and only the eigen values scale by the scalar multiplier. Hence we obtain the eigen-decomposition of $\tilde{A}$ as:

$$\tilde{\Lambda} = r_\xi \begin{pmatrix} u' - a & 0 & 0 & 0 \\ 0 & u' & 0 & 0 \\ 0 & 0 & u' + a & 0 \\ 0 & 0 & 0 & u' \end{pmatrix}$$

$$\tilde{R} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ u - a n_x & u & u + a n_x & n_y \\ v - a n_y & v & v + a n_y & -n_x \\ h_o - a u' & q & h_o + a u' & u n_y - v n_x \end{pmatrix}$$

$$\tilde{R}^{-1} = \begin{pmatrix} \frac{(\gamma-1)q + a u'}{2a^2} & \frac{(1-\gamma)u - a n_x}{2a^2} & \frac{(1-\gamma)v - a n_y}{2a^2} & \frac{(\gamma-1)}{2a^2} \\ \frac{a^2 - (\gamma-1)q}{a^2} & \frac{(\gamma-1)u}{a^2} & \frac{(\gamma-1)v}{a^2} & \frac{(1-\gamma)}{a^2} \\ \frac{(\gamma-1)q - a u'}{2a^2} & \frac{(1-\gamma)u + a n_x}{2a^2} & \frac{(1-\gamma)v + a n_y}{2a^2} & \frac{(\gamma-1)}{2a^2} \\ v n_x - u n_y & n_y & -n_x & 0 \end{pmatrix}$$

### 3.2.2 Final Expressions for the split vectors

For the vector $\tilde{F}$:

$$\tilde{F} = \tilde{F}^{+} + \tilde{F}^{-}$$
$$\tilde{F}^{+} = \tilde{R}\tilde{\Lambda}^{+}\tilde{R}^{-1}\tilde{U}$$
$$\tilde{F}^{-} = \tilde{R}\tilde{\Lambda}^{-}\tilde{R}^{-1}\tilde{U}$$

Define:

$$r_\xi = \sqrt{\xi_x^2 + \xi_y^2}$$
$$n_x = \frac{\xi_x}{r_\xi}$$
$$n_y = \frac{\xi_y}{r_\xi}$$
$$u' = un_x + v_ny$$
$$q = \frac{u^2 + v^2}{2}$$
$$h_o = q + \frac{a^2}{\gamma - 1}$$
$$a = \sqrt{\frac{\gamma P}{\rho}}$$

$$\tilde{\Lambda}^{+} = r_\xi \begin{pmatrix} \frac{u'-a+\sqrt{(u'-a)^2+\epsilon^2}}{2} & 0 & 0 & 0 \\ 0 & \frac{u'+\sqrt{u'^2+\epsilon^2}}{2} & 0 & 0 \\ 0 & 0 & \frac{u'+a+\sqrt{(u'+a)^2+\epsilon^2}}{2} & 0 \\ 0 & 0 & 0 & \frac{u'+\sqrt{u'^2+\epsilon^2}}{2} \end{pmatrix}$$

$$\tilde{\Lambda}^{-} = r_\xi \begin{pmatrix} \frac{u'-a-\sqrt{(u'-a)^2+\epsilon^2}}{2} & 0 & 0 & 0 \\ 0 & \frac{u'-\sqrt{u'^2+\epsilon^2}}{2} & 0 & 0 \\ 0 & 0 & \frac{u'+a-\sqrt{(u'+a)^2+\epsilon^2}}{2} & 0 \\ 0 & 0 & 0 & \frac{u'-\sqrt{u'^2+\epsilon^2}}{2} \end{pmatrix}$$

$$\tilde{R} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ u - an_x & u & u + an_x & n_y \\ v - an_y & v & v + an_y & -n_x \\ h_o - au' & q & h_o + au' & un_y - vn_x \end{pmatrix}$$

$$\tilde{R}^{-1} = \begin{pmatrix} \frac{(\gamma-1)q+au'}{2a^2} & \frac{(1-\gamma)u-an_x}{2a^2} & \frac{(1-\gamma)v-an_y}{2a^2} & \frac{(\gamma-1)}{2a^2} \\ \frac{a^2-(\gamma-1)q}{a^2} & \frac{(\gamma-1)u}{a^2} & \frac{(\gamma-1)v}{a^2} & \frac{(1-\gamma)}{a^2} \\ \frac{(\gamma-1)q-au'}{2a^2} & \frac{(1-\gamma)u+an_x}{2a^2} & \frac{(1-\gamma)v+an_y}{2a^2} & \frac{(\gamma-1)}{2a^2} \\ vn_x - un_y & n_y & -n_x & 0 \end{pmatrix}$$

For the vector $\tilde{G}$:

$$\tilde{G} = \tilde{G}^{+} + \tilde{G}^{-}$$

$$\tilde{G}^{+} = \tilde{R}\tilde{\Lambda}^{+}\tilde{R}^{-1}\tilde{U}$$

$$\tilde{G}^{-} = \tilde{R}\tilde{\Lambda}^{-}\tilde{R}^{-1}\tilde{U}$$

Define:

$$r_\eta = \sqrt{\eta_x^2 + \eta_y^2}$$

$$n_x = \frac{\eta_x}{r_\eta}$$

$$n_y = \frac{\eta_y}{r_\eta}$$

$$u' = un_x + v_n y$$

$$q = \frac{u^2 + v^2}{2}$$

$$h_o = q + \frac{a^2}{\gamma - 1}$$

$$a = \sqrt{\frac{\gamma P}{\rho}}$$

$$\tilde{\Lambda}^{+} = r_\eta \begin{pmatrix} \frac{u'-a+\sqrt{(u'-a)^2+\epsilon^2}}{2} & 0 & 0 & 0 \\ 0 & \frac{u'+\sqrt{u'^2+\epsilon^2}}{2} & 0 & 0 \\ 0 & 0 & \frac{u'+a+\sqrt{(u'+a)^2+\epsilon^2}}{2} & 0 \\ 0 & 0 & 0 & \frac{u'+\sqrt{u'^2+\epsilon^2}}{2} \end{pmatrix}$$

$$\tilde{\Lambda}^{-} = r_\eta \begin{pmatrix} \frac{u'-a-\sqrt{(u'-a)^2+\epsilon^2}}{2} & 0 & 0 & 0 \\ 0 & \frac{u'-\sqrt{u'^2+\epsilon^2}}{2} & 0 & 0 \\ 0 & 0 & \frac{u'+a-\sqrt{(u'+a)^2+\epsilon^2}}{2} & 0 \\ 0 & 0 & 0 & \frac{u'-\sqrt{u'^2+\epsilon^2}}{2} \end{pmatrix}$$

$$\tilde{R} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ u - an_x & u & u + an_x & n_y \\ v - an_y & v & v + an_y & -n_x \\ h_o - au' & q & h_o + au' & un_y - vn_x \end{pmatrix}$$

$$\tilde{R}^{-1} = \begin{pmatrix} \frac{(\gamma-1)q+au'}{2a^2} & \frac{(1-\gamma)u-an_x}{2a^2} & \frac{(1-\gamma)v-an_y}{2a^2} & \frac{(\gamma-1)}{2a^2} \\ \frac{a^2-(\gamma-1)q}{a^2} & \frac{(\gamma-1)u}{a^2} & \frac{(\gamma-1)v}{a^2} & \frac{(1-\gamma)}{a^2} \\ \frac{(\gamma-1)q-au'}{2a^2} & \frac{(1-\gamma)u+an_x}{2a^2} & \frac{(1-\gamma)v+an_y}{2a^2} & \frac{(\gamma-1)}{2a^2} \\ vn_x - un_y & n_y & -n_x & 0 \end{pmatrix}$$

### 3.3 Final Discretized Version

After the flux-splitting operation is performed, we can write the discrete for the Euler Equation. The original equation reads:

$$\frac{\partial \tilde{U}}{\partial t} + \frac{\partial \tilde{F}}{\partial \xi} + \frac{\partial \tilde{G}}{\partial \eta} = 0$$

Using Forward-Euler for time, first-order upwind for space:

$$\frac{\tilde{U}_{i,j}^{n+1} - \tilde{U}_{i,j}^{n}}{\Delta t} + \left(\frac{\tilde{F}_{i,j}^{+} - \tilde{F}_{i-1,j}^{+}}{\Delta \xi}\right)^n + \left(\frac{\tilde{F}_{i+1,j}^{-} - \tilde{F}_{i,j}^{-}}{\Delta \xi}\right)^n + \left(\frac{\tilde{G}_{i,j}^{+} - \tilde{G}_{i-1,j}^{+}}{\Delta \eta}\right)^n + \left(\frac{\tilde{G}_{i+1,j}^{-} - \tilde{G}_{i,j}^{-}}{\Delta \eta}\right)^n = 0$$

Let:

$$R_{i,j}^{n} = \left(\frac{\tilde{F}_{i,j}^{+} - \tilde{F}_{i-1,j}^{+}}{\Delta \xi}\right)^n + \left(\frac{\tilde{F}_{i+1,j}^{-} - \tilde{F}_{i,j}^{-}}{\Delta \xi}\right)^n + \left(\frac{\tilde{G}_{i,j}^{+} - \tilde{G}_{i,j-1}^{+}}{\Delta \eta}\right)^n + \left(\frac{\tilde{G}_{i,j+1}^{-} - \tilde{G}_{i,j}^{-}}{\Delta \eta}\right)^n$$

Hence:

$$\frac{\tilde{U}_{i,j}^{n+1} - \tilde{U}_{i,j}^{n}}{\Delta t} + R_{i,j}^{n} = 0$$

Dropping the $\tilde{}$ for $U$:

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n}}{\Delta t} + J_{i,j} R_{i,j}^{n} = 0$$

Hence the advance step is simply:

$$U_{i,j}^{n+1} = U_{i,j}^{n} - \Delta t J_{i,j} R_{i,j}^{n}$$

Where we note that $\Delta t$ must be chosen based on the max eigen-value:

$$\lambda_{i,j}^{max,\xi} = (|\xi_x u + \xi_y v| + \sqrt{\xi_x^2 + \xi_y^2} a)_{i,j}$$

$$\lambda_{i,j}^{max,\eta} = (|\eta_x u + \eta_y v| + \sqrt{\eta_x^2 + \eta_y^2} a)_{i,j}$$

$$\Delta t = \min_{\mathrm{i,j}} \left( \min \left( \frac{\Delta \xi}{\lambda_{\mathrm{i,j}}^{\mathrm{max},\xi}}, \frac{\Delta \eta}{\lambda_{\mathrm{i,j}}^{\mathrm{max},\eta}} \right) \right)$$

To be well inside the stable region:

$$\Delta t = CFL \times (\Delta t)_{min}$$

### 3.4 Boundary Conditions

We impose the periodic boundary condition along the $\xi-$ direction ($\eta-$constant lines) simply due to the fact that this trajectory is a closed loop (ellipse). Hence for any quantity $Q$, using the index-notation we follow:

$$Q(1, j) = Q(N, j)$$

The $\xi-$constant direction differs at the airfoil surface ($j = 1$) and the outer boundary $j = N$ and this is treated separately.

### 3.4.1 At the airfoil surface

and the slip-wall boundary condition at the airfoil surface for the $\eta-$direction. This is done by imposing:

$$\frac{\partial \tilde{u}}{\partial \eta} = 0, \tilde{v} = 0$$

Where:

$$\tilde{u} = \xi_x u + \xi_y v$$

$$\tilde{v} = \eta_x u + \eta_y v$$

Following the indexing notation used in this code, we note that the airfoil surface is accessed by setting $j = 1$. Looping through all the $i$s fetches the surface. Hence, the Neumann Boundary Condition at the airfoil surface can be written as:

$$\tilde{u}(i,1) = \tilde{u}(i,2)$$

Using the expression for $\tilde{u}$:

$$\xi_x(i,1)u(i,1) + \xi_y(i,1)v(i,1) = \xi_x(i,2)u(i,2) + \xi_y(i,2)v(i,2)$$

The red terms are unknown and need to be solved for. Similarly we write the Dirchlet BC to get another equation interms of the unknowns:

$$\tilde{v}(i,1) = \eta_x(i,1)u(i,1) + \eta_y(i,1)v(i,1) = 0$$

This can be written as a $2 \times 2$ system:

$$\begin{bmatrix} u(i,1) \\ v(i,1) \end{bmatrix} = \begin{pmatrix} \xi_x(i,1) & \xi_y(i,1) \\ \eta_x(i,1) & \eta_y(i,1) \end{pmatrix}^{-1} \begin{bmatrix} \xi_x(i,2)u(i,2) + \xi_y(i,2)v(i,2) \\ 0 \end{bmatrix}$$

Similarly:

$$\frac{\partial \rho}{\partial \eta} = 0, \frac{\partial(\rho E)}{\partial \eta} = 0$$

Following the same notation:

$$\rho(i,1) = \rho(i,2)$$

$$\rho E(i,1) = \rho E(i,2)$$

But we know that:

$$\rho E = \frac{1}{2}\rho(u^2 + v^2) + \frac{p}{\gamma - 1} = \rho q + \frac{p}{\gamma - 1}$$

Hence:

$$\rho(i,1)q(i,1) + \frac{p(i,1)}{\gamma - 1} = \rho(i,2)q(i,2) + \frac{p(i,2)}{\gamma - 1}$$

Hence the pressure at the airfoil surface can be obtained as:

$$p(i,1) = p(i,2) + (\gamma - 1)\left[\rho(i,2)q(i,2) - \rho(i,1)q(i,1)\right]$$

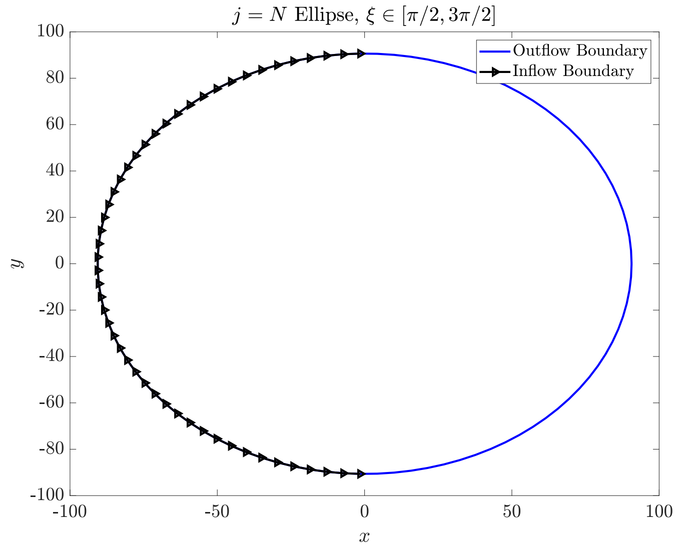



**Figure 4:** *Inflow and outflow segments of the outer boundary*

### 3.4.2 At the outer boundaries

For one half of the domain, we impose constant inflow (Dirichlet BC). We note that the $\xi-$coordinate is simply an angle that goes from 0 to $2\pi$ in the anti-clockwise sense. Hence, the inflow boundary is located using $\xi \in [\pi/2, 3\pi/2]$ as shown in Fig. 4 In terms of the index notation, this translates to moving between $i = N/4$ to $i = 3N/4$. Hence:

$$u\left(\frac{N}{4} \rightarrow \frac{3N}{4}, j\right) = u_{inflow}$$

$$v\left(\frac{N}{4} \rightarrow \frac{3N}{4}, j\right) = v_{inflow} = 0$$

$$p\left(\frac{N}{4} \rightarrow \frac{3N}{4}, j\right) = p_{inflow}$$

$$\rho\left(\frac{N}{4} \rightarrow \frac{3N}{4}, j\right) = \rho_{inflow}$$

and for the other half we perform outflow using:

$$\frac{\partial u}{\partial \eta} = 0, \frac{\partial v}{\partial \eta} = 0, \frac{\partial \rho}{\partial \eta} = 0, \frac{\partial (\rho E)}{\partial \eta} = 0$$

Hence:

$$u(i, N) = u(i, N-1)$$
$$v(i, N) = v(i, N-1)$$
$$\rho(i, N) = \rho(i, N-1)$$

We may write the last condition as:

$$\rho E(i, N) = \rho E(i, N-1)$$

Expanding and expressing for pressure:

$$p(i,N) = p(i,N-1) + (\gamma - 1)\left[\rho(i,N-1)q(i,N-1) - \rho(i,N)q(i,N)\right]$$

Strictly speaking, both the subsonic inflow and outflow boundaries need a charateristics-based treatment using the Navier-Stokes Characteristics Boundary Condition (NSCBC) approach. However, we circumvent these problems by choosing both the inflow and outflow surface very far from the airfoil surface. Since there's a lot of steps involved, we first write a pseudo-code to aid the code development process.

## 3.5 Pseudo-code

We use the index $i$ to traverse along the ellipses (constant-$\eta$ lines) and $j$ to traverse through the hyperbolas (constant-$\xi$ lines).

**Algorithm 1** Main solution loop

**while** $t < t_{max}$ **do**
  Compute $dt_{min}$
  $dt = CFL \times dt_{min}$
  **while** $i \in [1, N-1]$ **do**
    **while** $j \in [2, N-1]$ **do**
      (Do advance step $U_{i,j}^{new} = U_{i,j}^{old} - (\Delta t)_{min} J_{i,j} R_{i,j}$):
      First for $i = 1$ by imposing periodicity
      Then advance $i = 2 \to N-1$
      Set state($i = N$,j) = state($i = N$,j)
      Extract primitive variables from conserved solution variables
      Compute norms and store
    **end while**
  **end while**
  Advance airfoil surface $j = 1$
  Advance outer boundary $j = N$
  Update primitive vars at $j = 1, N$
  $t = t + (dt)$
  $Q^{old} = Q^{new}$
**end while**

## 3.6 Conservative to Primitive conversion

We can easily convert between conserved state vector $U$ to the primitve state vector $Q$:

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, Q = \begin{bmatrix} \rho \\ u \\ v \\ p \end{bmatrix}$$

Where:

$$\rho E = \frac{1}{2}\rho(u^2 + v^2) + \frac{p}{\gamma - 1}$$

Hence we first extract $u, v, rho$ using:

$$\rho = \rho, u = \frac{\rho u}{\rho}, v = \frac{\rho v}{\rho}$$

And then obtain pressure using:

$$p = (\gamma - 1)\left[(\rho E) - \frac{1}{2}(u^2 + v^2)\right]$$

## 3.7 Algorithm used to compute Drag Coefficient

We note that the pressure computed is normal to the airfoil/ellipse surface everywhere.

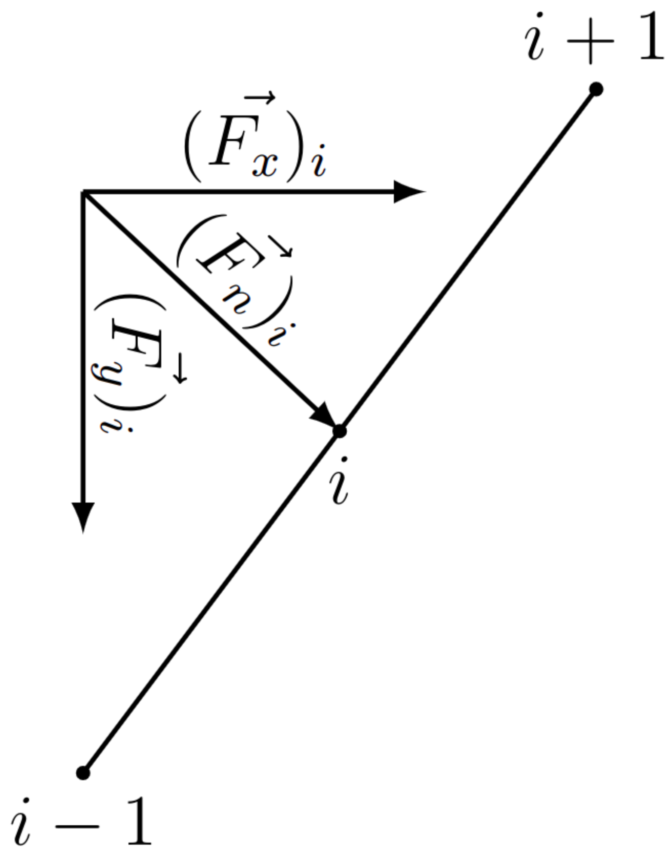



**Figure 5:** *Normal force at a location $i$ on the airfoil surface.*

Hence at a node $i$, we note that the local normal force can be approximated as

$$(F_n)_i = (pA)_i \approx p \times l_i = p_i\sqrt{(x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2}$$

Let $(F_x)_i$ and $(F_y)_i$ denote the $x$ and $y$ components of the local force. We note that there is no tangential force due to the pressure and thus we may write the following 2 equations in terms of $(F_x)_i$ and $(F_y)_i$(dropping the i's):

$$\xi_x F_x + \xi_y F_y = 0$$

$$\eta_x F_x + \eta_y F_y = F_n$$

hence:

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix}^{-1} \begin{bmatrix} 0 \\ F_n \end{bmatrix}$$

Solving for $(F_x)_i$ and summing over all $i's$ fetches the net drag force $F_D$. To obtain the drag coefficient, we use the definition:

$$C_D = \frac{F_D}{\frac{1}{2}\rho_\infty U_\infty^2 A}$$

Where $A$ here is the total area of the ellipse. In 2-D this reduces to just the total length of the ellipse that can be obtained by:

$$L = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

Hence the drag coefficient becomes:

$$C_D = \frac{F_D}{\frac{1}{2}\rho_\infty U_\infty L}$$

# 4 Results

## 4.1 Mesh Metrics

The Jacobian of space-transformation and the derivatives $\xi_x, \xi_y, \eta_x, \eta_y$ are plotted in Figs. 6 & 7.

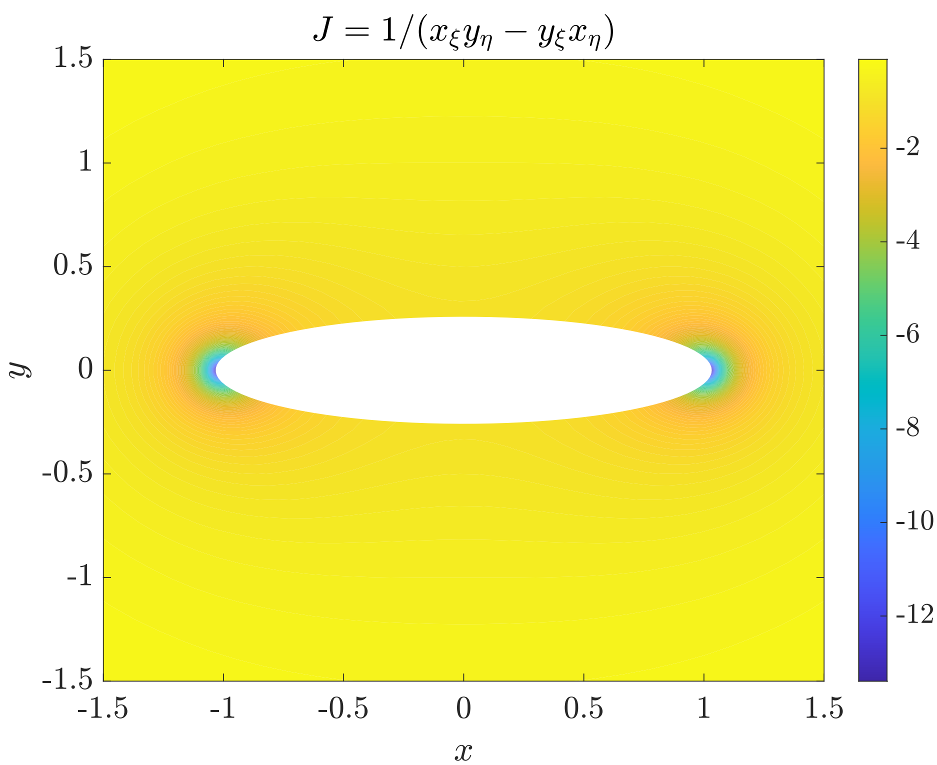



**Figure 6:** *Jacobian of transformation*

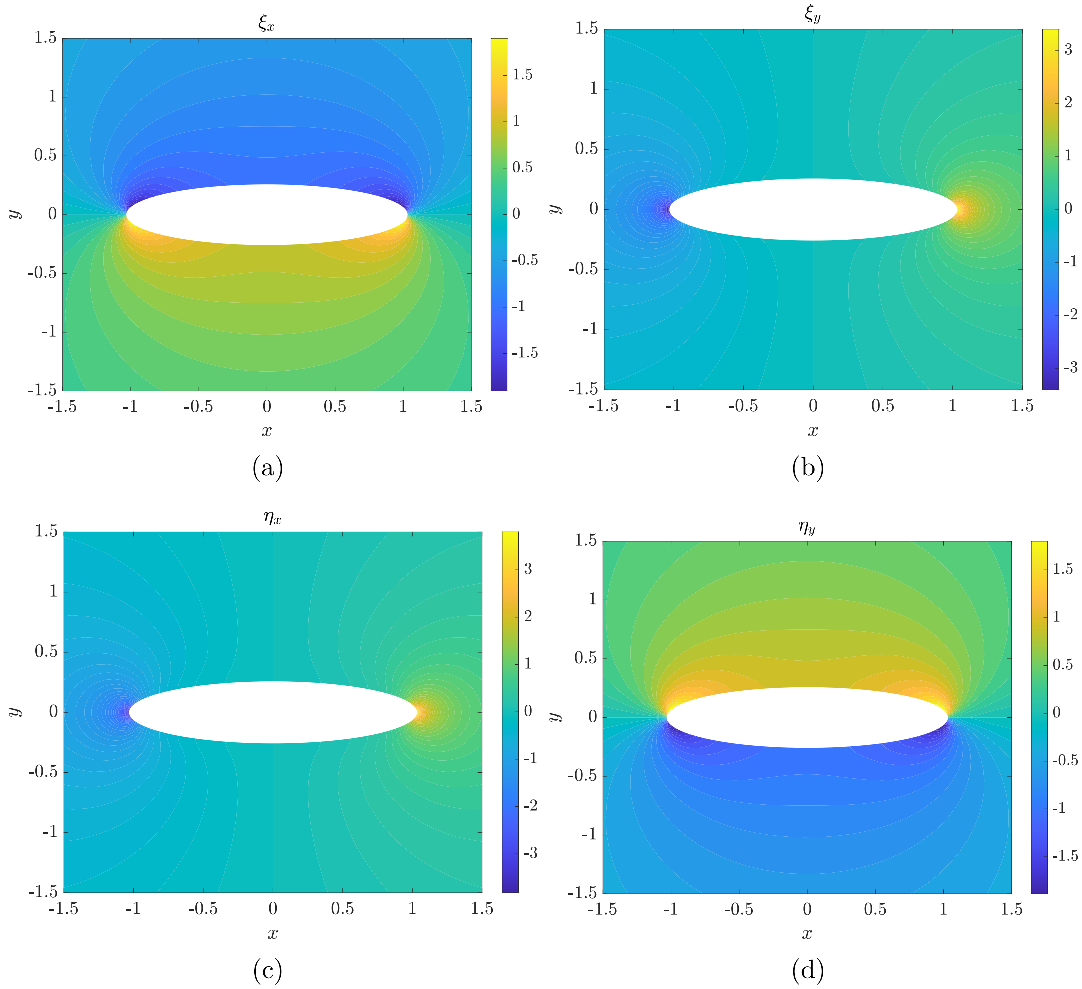



**Figure 7:** *Derivatives of transformation plotted in physical space*

## 4.2 Contour Plots and Residuals

For all cases, we run to $t = 10$ with an average $dt$ of approximately $10^{-3}$. Hence the average number of iterations run is about $10^4$. For each case, we plot the $u$, $v$, $|\bar{U}| = \sqrt{u^2 + v^2}$, $p$, $\rho$ fields in the physical space. Furthermore, we define a residual based on the $L_2$-norm for a quantity $Q$ as:

$$res = L_2(Q^{n+1} - Q^n) = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}(Q_{ij}^{n+1} - Q_{ij}^n)^2}$$

We plot the residuals vs. time to monitor the convergence of each of the primitive variables.

### 4.2.1 M=0.45

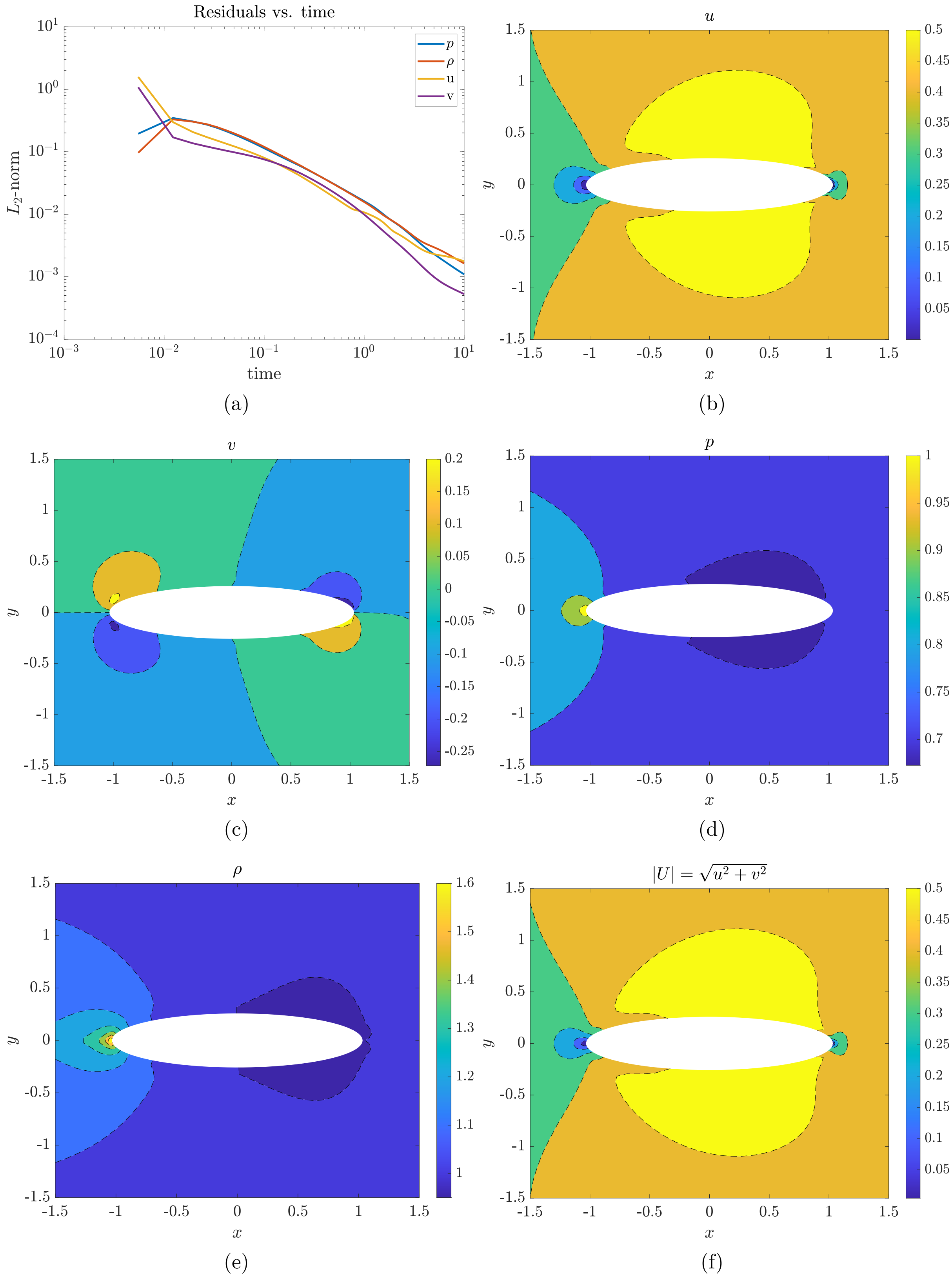



**Figure 8:** *Computed velocity, density and pressure fields for a subsonic case of* $M = 0.4$*.*

### 4.2.2 M=0.85

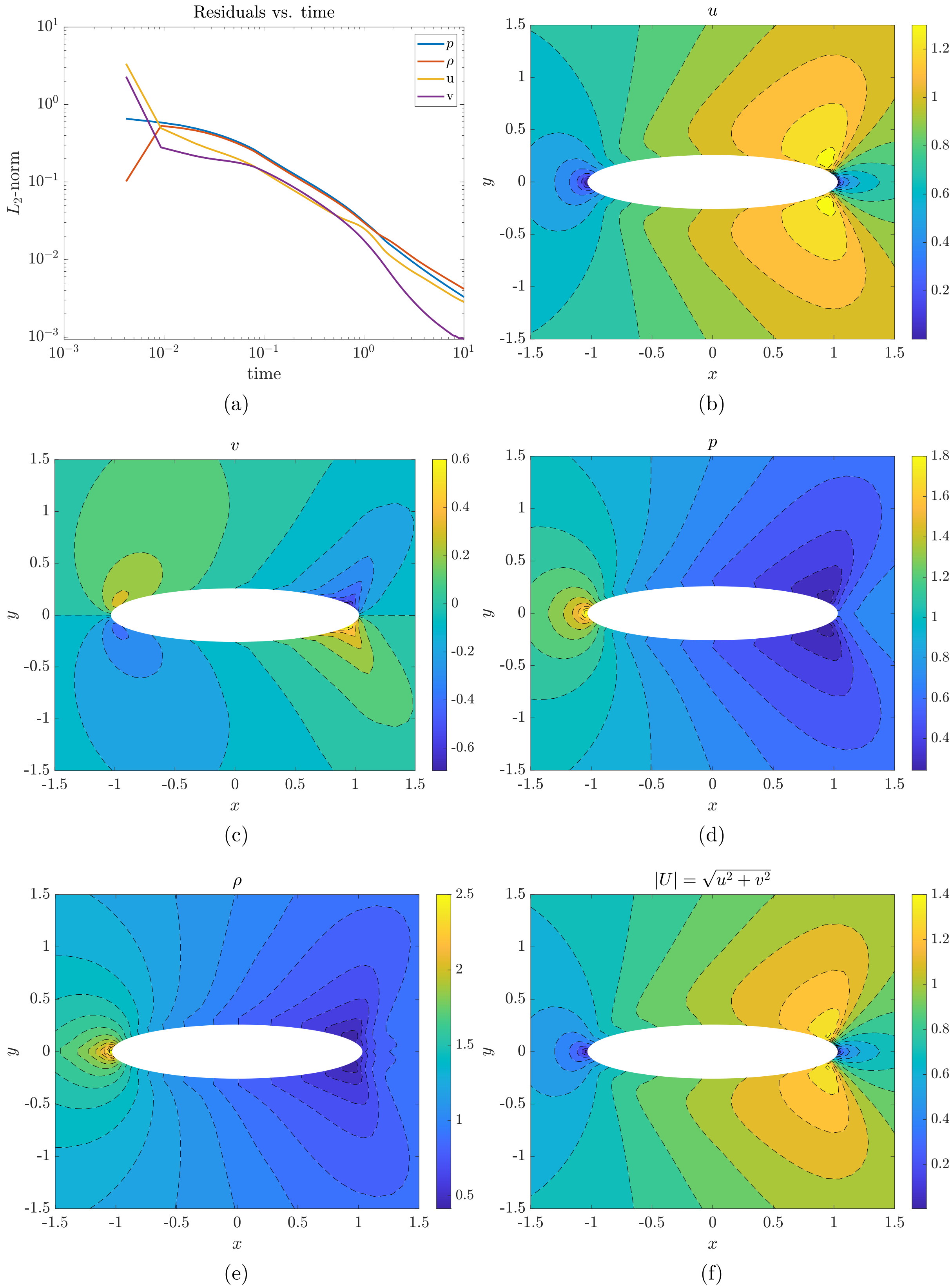



**Figure 9:** *Computed velocity, density and pressure fields for a subsonic case of* $M = 0.85$.

### 4.2.3 M=1.5

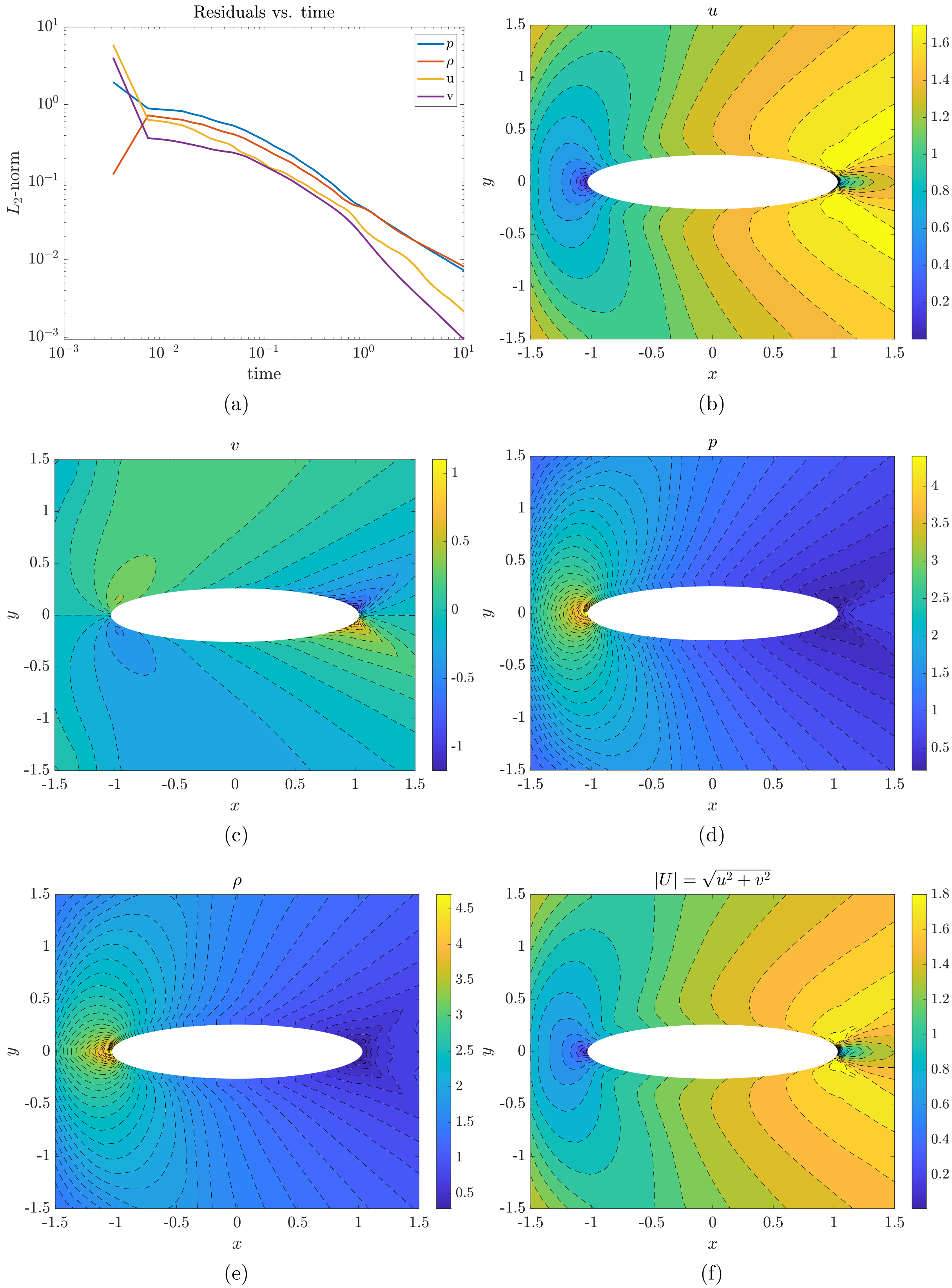



**Figure 10:** *Computed velocity, density and pressure fields for a subsonic case of $M = 1.5$.*

### 4.2.4 M=6.00

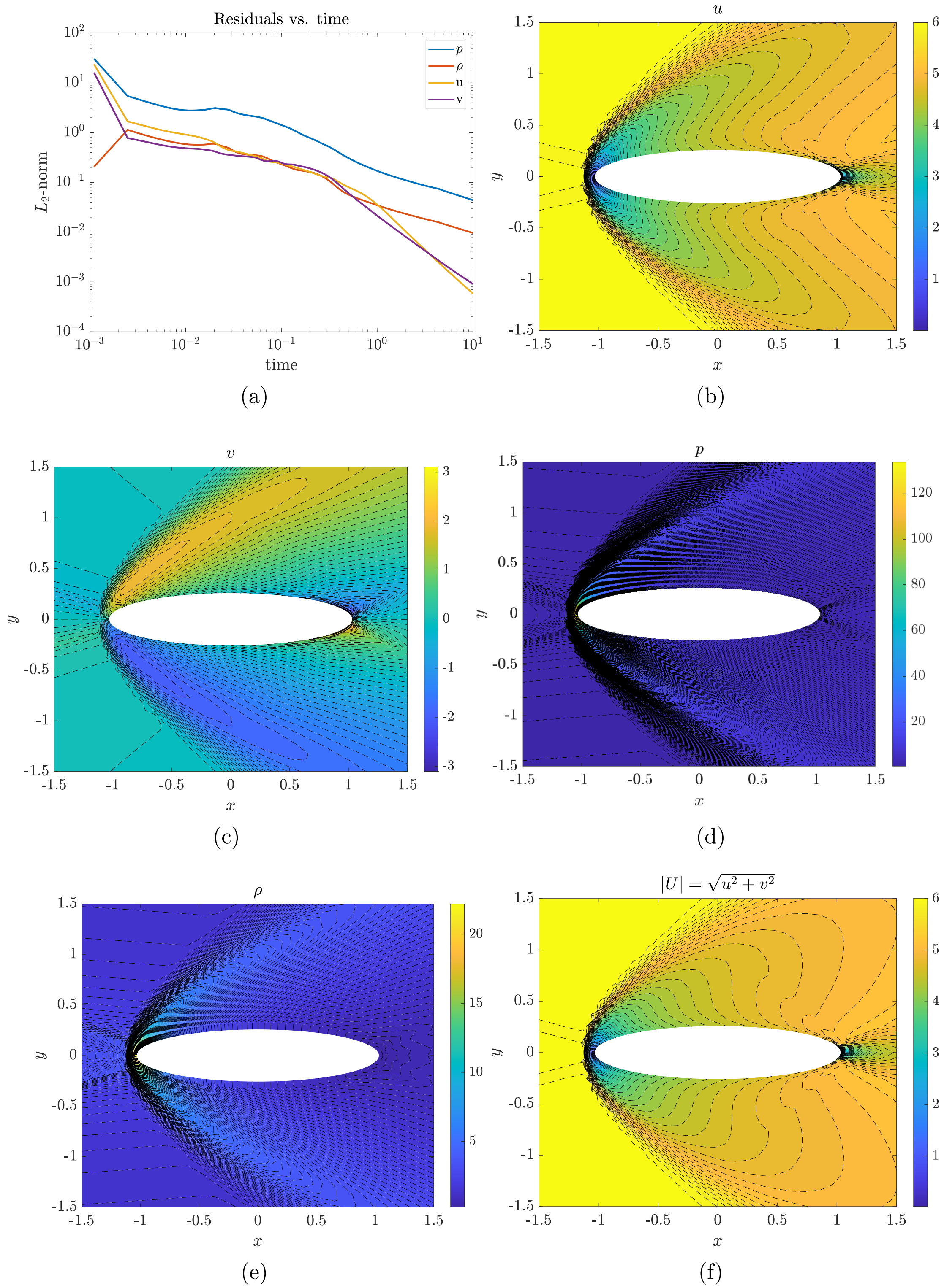



**Figure 11:** *Computed velocity, density and pressure fields for a subsonic case of* $M = 6.0$.

## 4.3 Discussion on Flow features

### 4.3.1 M=0.4

For this case, all regions are subsonic($M < 1$). We notice the formation of a region of high pressure at the ellipse's left tip which indicates the zone of stagnation. The velocity goes to zero here and the pressure goes to a maximum. We may approximate this stagnation pressure theoretically to be:

$$p_0 = p_\infty + \frac{1}{2}\rho_\infty U_\infty^2 = 0.7 + 0.5 \times 1 \times 0.45^2 \approx 0.8$$

We see that the contour plots also exhibit this value approximately.

### 4.3.2 M=0.85

We note that this is a transonic case ($M \approx 1$) where the mach number is close to one. We see that there is a formation of an oblique shock at approximately Mach 1 midway on the ellipse surface. The stagnation pressure rise is higher for this case compared to the previous case as expected. The subsonic regions exist upstream of this shock as visible from contours in Fig. 9.

### 4.3.3 M=1.5

This is a supersonic case ($M > 1$) and we note the distinct formation of a bow shock at the tip of the ellipse. The Mach number upstream of the shock is approximately validated using the normal shock relations. We use the online calculator developed by Virginia Tech for this purpose: https://devenport.aoe.vt.edu/aoe3114/calc.html and set $M_1 = 1.5$ in the normal shock section fetching us $M_2 = 0.7$. This is what is observed approximately in the $u$ contours shown in Fig. 10. The wake region of the ellipse is still subsonic with the mach numbers in the range $0.5 - 0.7$.

### 4.3.4 M=6.0

This is a hypersonic case ($M > 4$) for which we note that there is a greater difficulty in convergence. Residuals reach $10^{-3}$ despite performing $10^4$ iterations. Nonetheless, we observe the distinct formation of a strong bow shock with a smaller inclination angle with the horizontal compared to the previous case. The wake region is also supersonic here with the post-shock Mach number dropping to approximately 3-4. All regions are hence supersonic.

## 4.4 Drag Coefficients

We obtain the drag coefficients for each using the algorithm mentioned as follows:

| $\boldsymbol{M}$ | $\boldsymbol{C_D}$ |
|---|---|
| 0.4 | 0.81 |
| 0.85 | 1.07 |
| 1.5 | 0.98 |
| 6.00 | 0.12 |

We plot these results on a graph in Fig. 12 to see that the $C_D$ value peaks at $M = 0.85$ and drops at subsequent mach numbers demonstrating the "sound-barrier". which is the increase of the induced drag as the Mach number approaches 1.

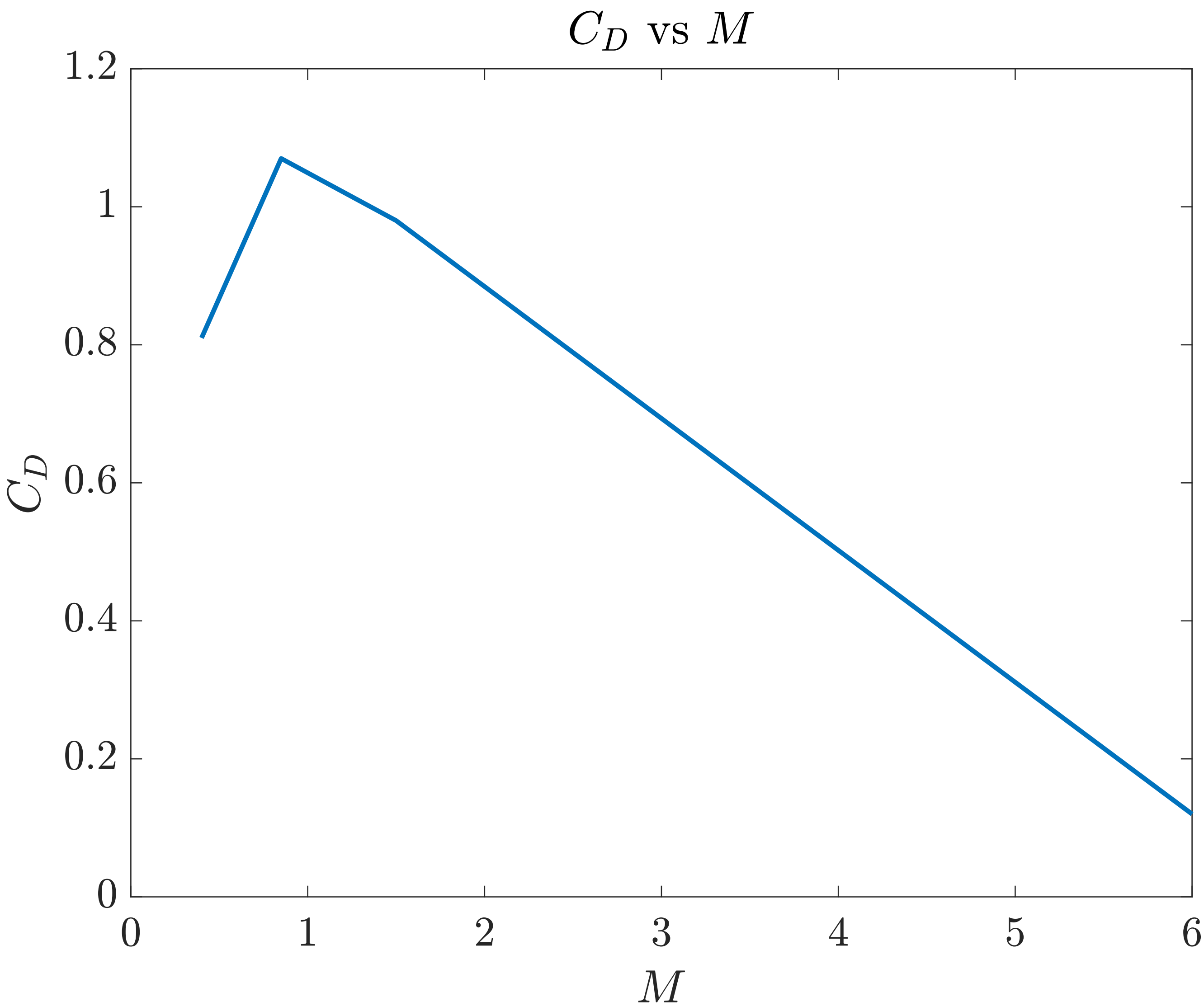



**Figure 12:** *$C_D$ vs. $M$ illustrating the "sound-barrier". Peak drag is observed as we approach Mach 1.*

## References

[1] P Buning and J Steger. "Solution of the two-dimensional Euler equations with generalized coordinate transformation using flux vector splitting". In: *3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference*. 1982, p. 971.