Numerical Study of Possion Equation in Subsonic and Supersonic Airfoil Flow

Liu Deng

School of Astronautics, Xi'an Jiaotong University, Xi'an, 710049, China

17267855816@stu.xjtu.edu.cn

Abstract. In the context of subsonic and supersonic airfoil flows in fluid mechanics, this study aims to numerically simulate the Poisson equation. The focus is on analyzing the variation of fluid flow under stable conditions of incompressible fluid. The research classifies the flow into four distinct cases based on the Mach number and its constancy and employs Python for intricate numerical computation. The culmination of the study involves generating Mach number contour plots and streamlining diagrams for each case using Python, which visually depicts the nuanced evolution of the flow field. Moreover, the study conducts a rigorous comparative analysis of the simulation results for the different cases, exploring the intricate differences in flow behavior across a spectrum of Mach numbers and flow conditions. The discussion section further broadens the scope of inquiry by addressing potential limitations and charting out prospective research directions. By providing a comprehensive exploration of the widespread application of the Poisson equation in fluid mechanics, this paper stands as a beacon for further investigations in the domain and presents invaluable insights for the progressive advancement of fluid mechanics studies in the future.

Keywords: Poisson equation, numerical simulation, subsonic airfoil flows, supersonic airfoil flows, incompressible fluid.

1. Introduction

With various numerical calculation methods and industrial simulation software development, the study of aerodynamic analysis has made corresponding progress. Aviation designers can optimize and improve the shape design of aircraft through dynamic analysis to improve the aircraft's flight speed and save more costs for airlines. It can also reduce time connections and improve business and international relations [1]. The commonly used aerodynamic analysis methods include wind tunnel experiments and numerical simulation research (CFD). As the most realistic physical flow simulation method before the flight test, the wind tunnel experiment can verify the aerodynamic design and be the main way to obtain an aircraft aerodynamic database. However, there are some problems in wind tunnel experiments, such as high costs and increasing demand for special test technology [2]. In contrast, computational fluid dynamics has the characteristics of low cost, relatively high accuracy, and can reduce the number of tasks in wind tunnel experiments. Scientists and engineers work by writing code that solves a common equation for studying the flow around an aircraft. These differential equations took time and effort before computers [3-5].

The current development of aerodynamic analysis methods mainly includes three requirements: accuracy, complexity processing, and visibility. Accuracy is crucial in the aerodynamic analysis of aircraft, as even small errors can significantly impact the vehicle's performance. For example, Radulescu considered the effects of Angle of attack, wind speed, air viscosity and shape curvature when studying the solution of optimizing the shape of airfoil into supersonic viscous fluid so that the obtained results could be better applied to supersonic flow [6]. Complexity is also necessary. The wing usually has a complex three-dimensional shape and a variety of airfoils, which will have a complex impact on the aerodynamic performance of the wing surface, so it is sometimes necessary to simplify the problem according to the situation [7]. When the airfoil is very thin, some researchers will use the computational fluid dynamics method to deal with the aerodynamic analysis of the wing as a two-dimensional problem [8, 9]. Compared to three-dimensional problems, it can improve efficiency and speed. The requirements for aerodynamic analysis methods also include the ability to

provide visualizations so that scientists and engineers can intuitively understand the results for analysis and decision making. For example, Liu et al. listed the wavelet coefficient cloud image corresponding to the leading edge shear layer in various working conditions when studying the bionic feathering control speed flow of fixed-wing UAV to better analyze the conclusion that the flexible feathering installed at the trailing edge has excellent stalling flow control effect compared with the leading edge [10].

In this paper, the numerical solution of the velocity potential function corresponding to the two-dimensional Poisson equation at subsonic and supersonic speeds is studied, simplifying the problem on the premise that the conditions are as close as possible to the real situation.

2. Derivation of Discretized Poisson Equation

2.1. Derivation of Poisson Equation

Assuming that vorticity is 0. The 3D flow of steady fluid is described by a simplified continuity equation (2). Under the condition of constant entropy, it is possible to derive the equation (3)

$$\boldsymbol{v} = \nabla\phi \tag{1}$$

$$div(\rho\nabla\phi) = 0 \tag{2}$$

$$\frac{\rho}{\rho_0} = \left(1 - \frac{|\nabla\phi|^2}{2H_0} - \frac{\partial_t\phi}{H_0}\right)^{1/(\gamma-1)} \tag{3}$$

In the above equations, ρ_0 is the stagnation density. H_0 is the stagnation enthalpy. By simplifying, a single scalar equation that describes the fluid flow can be led to:

$$(1 - M_x^2)\phi_{xx} + (1 - M_y^2)\phi_{yy} + (1 - M_z^2)\phi_{zz} - 2M_xM_y\phi_{xy} - 2M_xM_z\phi_{xz} - 2M_yM_z\phi_{yz} = \frac{1}{c^2}[\phi_{tt} + \partial_t(|\nabla\phi|^2)] \tag{4}$$

M is the Mach number, and c is the speed of sound. The relationship of the variables in the above equation is as follows:

$$M_x = \frac{\phi_x}{c}\ M_y = \frac{\phi_y}{c}\ M_z = \frac{\phi_z}{c} \tag{5}$$

$$c^2 = (\gamma - 1)\left[H_0 - \frac{1}{2}|\nabla\phi|^2 - \phi_t\right] \tag{6}$$

The primary focus of this article is on steady flow, and therefore all-time derivatives are omitted:

$$(1 - M_x^2)\phi_{xx} + (1 - M_y^2)\phi_{yy} + (1 - M_z^2)\phi_{zz} - 2M_xM_y\phi_{xy} - 2\mathrm{M_xM_z}\phi_{\mathrm{xz}} - 2M_yM_z\phi_{yz} = 0 \tag{7}$$

In the case of thin obstacles, such as thin airfoils, the discussion will be confined to a 2D plane. Therefore the simplification of the full potential equation is done.

Considering the airfoil has a small effect on the flow, the perturbation to flow with the velocity of magnitude U_∞ in the x-direction is considered. The potential has a representation:

$$\phi = U_\infty(x + \Phi) \tag{8}$$

Velocities are recovered from the potential:

$$u = U_\infty(1 + \Phi_x) \tag{9}$$

$$\mathrm{v} = \mathrm{U}_\infty\Phi_\mathrm{y} \tag{10}$$

The equation becomes:

$$(1 - M_x^2)\Phi_{xx} + \Phi_{yy} = 0 \tag{11}$$

Which can be further simplified into the following:

$$(1 - M_\infty^2)\Phi_{xx} + \Phi_{yy} = 0 \tag{12}$$

2.2. When the Mach Number Is Constant

The equation that we need to solve is $(1-M_\infty^2)\varphi_{xx}+\varphi_{yy}=0$. First, it is necessary to discretize the given differential equation. The discretization method is different when Ma_∞ is less than 1, and when Ma_∞ is greater than 1. So both cases will be discussed. It is assumed that $\triangle x = \triangle y$.

When Ma_∞ is less than 1, the following holds:

$$(1-M_\infty^2)\left(\varphi_{i+1,j}-2\varphi_{i,j}+\varphi_{i-1,j}\right)+\left(\varphi_{i,j+1}-2\varphi_{i,j}+\varphi_{i,j-1}\right)=0 \tag{13}$$

Through integration, the equation above becomes:

$$-(4-2M_\infty^2)\varphi_{i,j}+(1-M_\infty^2)\varphi_{i+1,j}+(1-M_\infty^2)\varphi_{i-1,j}+\varphi_{i,j-1}+\varphi_{i,j+1}=0 \tag{14}$$

When Ma_∞ is larger than 1, the following holds:

$$(1-M_\infty^2)\left(\varphi_{i,j}-2\varphi_{i-1,j}+\varphi_{i-2,j}\right)+\left(\varphi_{i,j+1}-2\varphi_{i,j}+\varphi_{i,j-1}\right)=0 \tag{15}$$

Through integration, the equation above becomes:

$$-(1+M_\infty^2)\varphi_{i,j}-2(1-M_\infty^2)\varphi_{i-1,j}+(1-M_\infty^2)\varphi_{i-2,j}+\varphi_{i,j-1}+\varphi_{i,j+1}=0 \tag{16}$$

2.3. When the Mach Number Is a Variable

Under the condition that the Mach number is variable, equation 12 can be written as:

$$[1-M_\infty^2-(\gamma+1)M_\infty^2\varphi_x]\varphi_{xx}+\varphi_{yy}=0 \tag{17}$$

In this equation, $K_{i,j}=1+M_\infty^2-(\gamma+1)M_\infty^2\frac{(\varphi_{i+1,j}-\varphi_{i-1,j})}{2\triangle x}$

For $Ma_\infty<1$,

$$K_{i,j}\left(\varphi_{i+1,j}-2\varphi_{i,j}+\varphi_{i-1,j}\right)+\left(\varphi_{i,j+1}-2\varphi_{i,j}+\varphi_{i,j-1}\right)=0 \tag{18}$$

Through integration:

$$-\left(2K_{i,j}+2\right)\varphi_{i,j}+K_{i,j}\varphi_{i+1,j}+K_{i,j}\varphi_{i-1,j}+\varphi_{i,j-1}+\varphi_{i,j+1}=0 \tag{19}$$

For $Ma_\infty>1$,

$$K_{i-1,j}\left(\varphi_{i,j}-2\varphi_{i-1,j}+\varphi_{i-2,j}\right)+\left(\varphi_{i,j+1}-2\varphi_{i,j}+\varphi_{i,j-1}\right)=0 \tag{20}$$

Through integration:

$$-\left(2-K_{i-1,j}\right)\varphi_{i,j}-2K_{i-1,j}\varphi_{i-1,j}+K_{i-1,j}\varphi_{i-2,j}+\varphi_{i,j-1}+\varphi_{i,j+1}=0 \tag{21}$$

3. The Boundary Conditions

The equation that needs to be solved is defined in the domain $[0,3]\times[0,1]$. The code is written to apply the small disturbance BC at the bottom boundary (Fig. 1):

$$-\frac{\partial\phi}{\partial x}=U_\infty f'(x) \tag{22}$$

Where $f(x)$ is the shape of the airfoil defined on $x\in[1,2]$. Use for

$$f(x)=\beta\sin\left(\pi(x-1)\right) \tag{23}$$

```
def BC_F(x):
    pi = np.pi
    beta=0.1
    if x < 1 or x > 2:
        return 0.0
    else:
        return U_inf*beta * pi*np.cos(pi * (x - 1))
```

Figure 1. Functions that define boundary conditions (Photo/Picture credit: Original)

Equation (22) can be discretized to:

$$\frac{1}{2h}\left[\varphi_{i,j+1} - \varphi_{i,j-1}\right] = BC_F(ih) \tag{24}$$

From which the most value $\varphi_{i,j-1}$ can be represented by $\varphi_{i,j+1}$ and boundary conditions at j=0.

4. Solving the Poisson Equation in Subsonic Flow

In Python code, four functions are set. The BC-F function defines the Neumann boundary condition. The velocity-vector function defines the velocity in both the x and y directions. The error function is defined to calculate the error, which evaluates the accuracy of our computational results. The SOR function is defined for iterative computation, mainly used in the successive over-relaxation method. It aids in performing iterative computations effectively to achieve the ultimate numerical solution. After defining the function, the sizes of some constant variables are determined (Fig. 2):

```
N = 32
M_inf = 1.3
ho=1
r=1.4
U_inf = np.sqrt((np.square(M_inf)*ho*(r-1))/(1+0.5*np.square(M_inf)*(r-1)))
h = 1 / N
```

Figure 2. The value of constant variables (Photo/Picture credit: Original)

In the case of an airplane traveling at subsonic speeds, the coefficients of the operator can be described by a 5-point stencil, as shown in Fig. 3:

	B_{yp} $\varphi_{i,j+1}$	
A_{xm} $\varphi_{i-1,j}$	$-C_0$ $\varphi_{i,j}$	A_{xp} $\varphi_{i+1,j}$
	B_{ym} $\varphi_{i,j-1}$	

$$(1 - M^2) \cdot \varphi_{xx} + \varphi_{yy} = 0$$
$$= -C_0 \cdot \boldsymbol{\varphi_{i,j}} + A_{xm} \cdot \boldsymbol{\varphi_{i-1,j}} + A_{xp}$$
$$\cdot \boldsymbol{\varphi_{i+1,j}} + \boldsymbol{B_{ym}} \cdot \boldsymbol{\varphi_{i,j-1}} + \boldsymbol{B_{yp}}$$
$$\cdot \boldsymbol{\varphi_{i,j+1}}$$

Figure 3. 5-point stencil, $Ma_\infty < 1$ (Photo/Picture credit: Original)

4.1. When the Mach Number Is Constant

When the Mach number is less than 1, equation (14) is used. The error function and the SOR function are defined in Figs. 4 and 5:

```
def error(phi):
    newphi = phi.copy()
    Nx = newphi.shape[0]
    Ny = newphi.shape[1]
    err = 0.0
    for i in range(1, Nx - 1):
        for j in range(1, Ny - 1):
            A_xp = 1-M_inf**2
            A_xm = 1-M_inf**2
            B_yp = 1
            B_ym = 1
            C0 = (A_xp + A_xm + B_yp + B_ym)
            eq = - C0 * newphi[i, j] + (B_yp * newphi[i, j + 1] + B_ym * newphi[i, j - 1] +
                                        A_xp * newphi[i + 1, j] + A_xm * newphi[i - 1, j])
            err += abs(eq)
    return err / (Nx * Ny)
```

Figure 4. Error function, $Ma_\infty = 0.3$ (Photo/Picture credit: Original)

```
def SOR(phi):
    h = 3.0 / phi.shape[0]
    newphi = phi.copy()
    sor = 1.0
    for i in range(1, newphi.shape[0] - 1):
        for j in range(1, newphi.shape[1] - 1):
            A_xp = 1-M_inf**2
            A_xm = 1-M_inf**2
            B_yp = 1
            B_ym = 1
            C0 = (A_xp + A_xm + B_yp + B_ym)
            dp = -newphi[i, j] + (B_yp * newphi[i, j + 1] + B_ym * newphi[i, j - 1] +
                                  A_xp * newphi[i + 1, j] + A_xm * newphi[i - 1, j]) / C0
            newphi[i, j] += sor * dp
        j = 0
        for i in range(1, newphi.shape[0] - 1):
            A_xp = 1-M_inf**2
            A_xm = 1-M_inf**2
            B_yp = 1
            B_ym = 1
            C0 = (A_xp + A_xm + B_yp + B_ym)
            dp = -newphi[i, j] + (2 * B_yp * newphi[i, j + 1] +
                                  A_xp * newphi[i + 1, j] + A_xm * newphi[i - 1, j]
                                  - 2 * h * BC_F(i * h)) / C0
            newphi[i, j] += sor * dp
    return newphi
```

Figure 5. SOR function, $Ma_\infty = 0.3$ (Photo/Picture credit: Original)

4.2. When the Mach Number Is a Variable

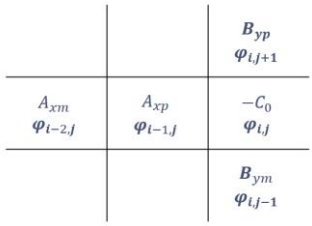
When the Mach number is less than 1, and the Mach number is not fixed, the only difference from the previous case is that instead of being $1 - M_\infty^2$, A_xp and A_xm are K[i, j]. The definition of K[i, j] is shown in Fig. 6:

```
for i in range (1,newphi.shape[0]-1):
    for j in range (0,newphi.shape[1]):
        k[i,j]=1-M_inf * M_inf - (r+1)*M_inf * M_inf*(newphi[i+1,j]
               -newphi[i-1,j])/(2*h)
```

Figure 6. The definition of K[i, j] (Photo/Picture credit: Original)

5. Solving the Poisson Equation in Supersonic Flow

In the case of an airplane traveling at supersonic speeds, the coefficients of the operator can be described by a 5-point stencil (Fig. 7):



$$(1 - M^2) \cdot \varphi_{xx} + \varphi_{yy} = 0$$
$$= -C_0 \cdot \boldsymbol{\varphi_{i,j}} + A_{xm} \cdot \boldsymbol{\varphi_{i-2,j}} - 2$$
$$\cdot A_{xp} \cdot \boldsymbol{\varphi_{i-1,j}} + \boldsymbol{B_{ym}} \cdot \boldsymbol{\varphi_{i,j-1}} + \boldsymbol{B_{yp}}$$
$$\cdot \boldsymbol{\varphi_{i,j+1}}$$

Figure 7. 5-point stencil, $Ma_\infty > 1$ (Photo/Picture credit: Original)

5.1. When the Mach Number Is Constant

Instead of a subsonic with a fixed Mach number, supersonic Mach becomes 1.3. The equation (16) is used, and the range of i changes accordingly.

5.2. When the Mach Number Is a Variable

In this case, all that must be done is to replace A_xp and A_xm in the previous example with K[i, j]. The SOR function is defined in Fig. 8:

```
def SOR(phi):
    h = 3.0 / phi.shape[0]
    newphi = phi.copy()
    sor = 1.0
    for i in range(2, newphi.shape[0]):
        for j in range(1, newphi.shape[1] -1):
            A_xp = 1-M_inf**2
            A_xm = 1-M_inf**2
            B_yp = 1
            B_ym = 1
            C0 = -(A_xp - B_yp - B_ym)
            dp= - newphi[i, j] + (B_yp * newphi[i, j + 1] + B_ym * newphi[i, j - 1]
             - A_xp * newphi[i - 1, j] + A_xm * newphi[i - 2, j]
             - A_xm * newphi[i - 1, j])/C0
            newphi[i, j] += sor * dp
        j = 0
        for i in range(2, newphi.shape[0]):
            A_xp = 1-M_inf**2
            A_xm = 1-M_inf**2
            B_yp = 1
            B_ym = 1
            C0 = -(A_xp -B_yp -B_ym)
            dp = -newphi[i, j] +(2 * B_yp * newphi[i, j + 1] -
                                 2* A_xp * newphi[i-1, j] + A_xm * newphi[i -2, j]
                                 - 2 * h * BC_F(i * h)) / C0
            newphi[i, j] += sor * dp
    return newphi
```

Figure 8. SOR function, $Ma_\infty = 1.3$ (Photo/Picture credit: Original)

6. Results

The operation results at subsonic speeds are shown in Figs. 9-14. The results of the operation at supersonic speed are shown in Figs. 11-16.

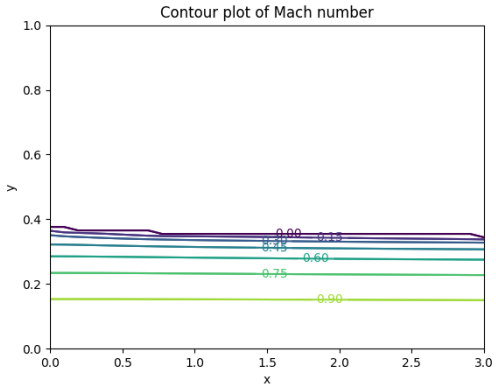


Figure 9. Contour plot of Mach number, Ma_∞=0.3 and is fixed (Photo/Picture credit: Original)

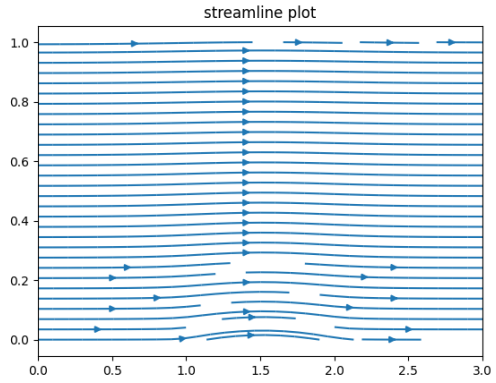


Figure 10. Streamline plot, Ma_∞=0.3 and is fixed (Photo/Picture credit: Original)

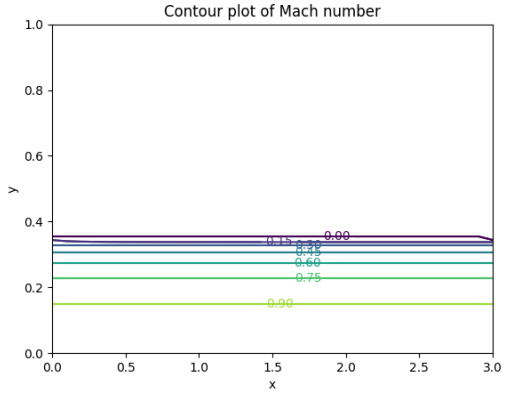


Figure 11. Contour plot of Mach number, Ma_∞=1.3 and is fixed (Photo/Picture credit: Original)

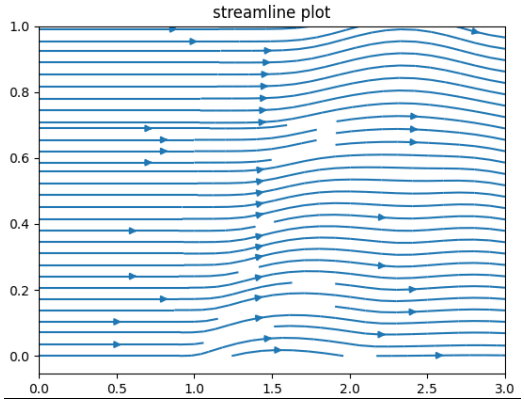


Figure 12. Streamline plot, Ma_∞=1.3 and is fixed (Photo/Picture credit: Original)

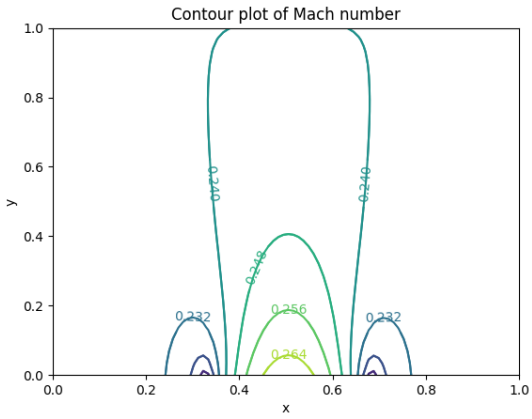


Figure 13. Contour plot of Mach number, Ma_∞=0.3, and K is not fixed (Photo/Picture credit: Original)

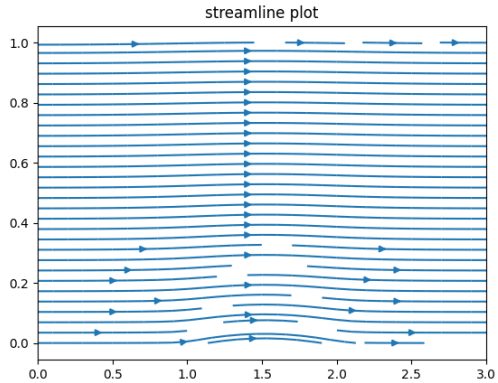


Figure 14. Streamline plot, Ma_∞=0.3, and K is not fixed (Photo/Picture credit: Original)

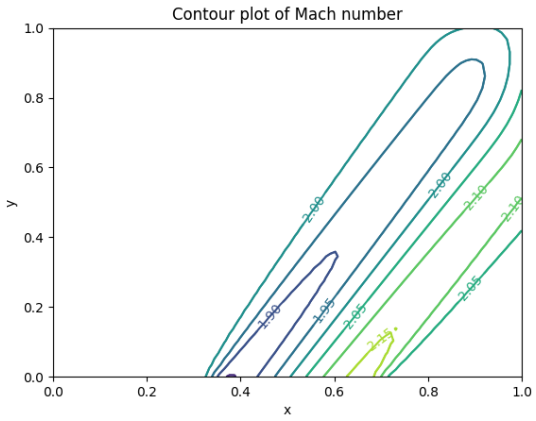


Figure 15. Contour plot of Mach number, Ma_∞=1.3 and K is not fixed (Photo/Picture credit: Original)

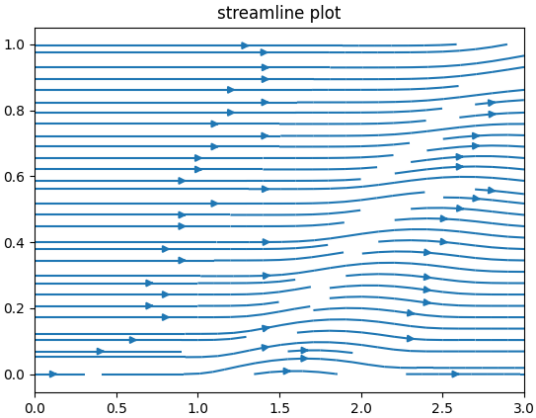


Figure 16. Streamline plot, Ma_∞=1.3 and K is not fixed (Photo/Picture credit: Original)

7. Conclusion

For a fluid with a constant Mach number, it can be seen from the comparison of Fig. 9 and Fig. 11 that there is almost a horizontal line in the contour diagram of Mach number, which means that the velocity distribution of the fluid in the vertical direction is uniform and there is no vertical velocity gradient change, which also implies the laminar flow characteristics and stable flow conditions of the fluid. The convection chart analysis can be found in Fig. 10 and Fig. 12. When the flow line in Fig. 10 becomes stable, and the horizontal region with the Mach number contour can deform in Fig. 12, this may be due to the change in fluid properties caused by shock waves, which implies the existence of shock waves. At this time, the conclusions about supersonic flow may need further analysis and verification.

For fluid with an unstable Mach number, it can be seen that the flow diagram is relatively stable during subsonic flow, and there is no shock wave reflection or deformation in the flow diagram. However, the supersonic flow analysis shows that the Mach number's contour diagram is curved and deformed, and the flow diagram is also curved, which indicates the presence of shock waves, and the two are in good agreement. Transonic flows can be extended based on existing supersonic and subsonic codes or optimized by adding the influence of turbulence factors. In the future, we will optimize the existing codes and determine the critical Mach number of turbulence based on flow velocity, geometry, flow stability, etc., to conduct more detailed studies.

References

[1] Aprovitola Andrea, Dyblenko Oleksandr and Pezzella Giuseppe, et al. Aerodynamic analysis of a supersonic transport aircraft at low and high speed flow conditions. Aerospace, 2022, 9 (8): 411.

[2] Chen Yingchun, Guo Chuanliang, Li Xiaoyong. Wind tunnel testing requirements and technology prospects for large commercial aircraft. Aerodynamic Research & Experiment, 2020, 32 (1): 1.

[3] Martínez Martí Enrique G. Flow and entropy characteristics around airfoils in subsonic and supersonic flows. University of Puerto Rico MayagÜez Campus, 2006.

[4] Claire Chainais-Hillairet, Yue-Jue Peng and Ingrid Violet. Numerical solutions of Euler–Poisson systems for potential flows. Applied Numerical Mathematics, 2009, 59 (2): 301-315.

[5] Toufik Zebbiche, Merouane Salhi and Mohamed Boun-djad. Numerical computation of supersonic flow around appointed airfoils. Journal of Computational Methods in Sciences and Engineering, 2012, 12 (3): 213-233.

[6] Victorita Radulescu. Solution to optimize the airfoils shapes placed into a supersonic viscous flow. In ASME International Mechanical Engineering Congress and Exposition, 2018, 52002: V001T03A001.

[7] Belova O. N., Kokoshinskaya N. S. and Paskonov V. M. Simplified navier-stokes equations for computing viscous gas flow past long bodies. Computational Mathematics and Modeling, 1990, 1 (1): 117-122.

[8] Alireza Karbalaei and Kazem Hejranfar. A central difference finite volume lattice Boltzmann method for simulation of 2d inviscid compressible flows on triangular meshes. In ASME International Mechanical Engineering Congress and Exposition, 2018, 52101: V007T09A025.

[9] Kryštůfek Pavel and Karel Kozel. Numerical solution of invisicid compressible steady flows around the RAE 2822 airfoil. In EPJ Web of Conferences, 2015, 92: 02041.

[10] Liu Yihong, Ma Xingxu, Gong Xuan, et al. Wind tunnel experiment on control of stalling flow of fixed wing UAV with bionic covert. Acta Aerodynamica Sinica, 2023, 41: 1-9.