

《计算空气动力学》 大作业

SX1501021 仓宇

2016 年 7 月 12 日

目录

1	问题描述	2
2	问题分析	2
2.1	基本方程	2
2.2	空间离散	3
2.3	人工耗散	4
2.4	时间离散	5
2.5	边界条件	5
3	编程实现	6
3.1	网格文件说明	6
3.2	数据存储结构	7
3.3	程序流程	8
4	结果分析	9
4.1	Ma=0.3	9
4.2	Ma=0.8	11
4.3	Ma=1.2	13
5	总结	15

1 问题描述

求解无粘条件下NACA0012翼型的2维平面流场。流场网格是由三角形单元组成的非结构网格，使用有限体积方法求解流场的2维Euler方程。网格文件为data文件夹下的naca0012.grd文件。全流场的网格如下左图所示，右图是翼型周围的网格：

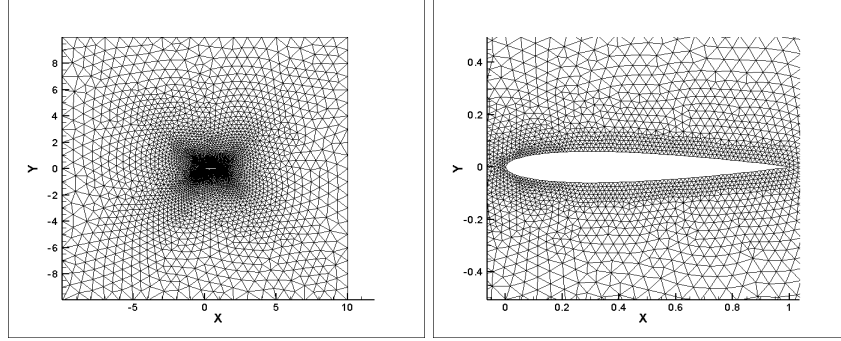


图 1: 用于NACA0012翼型的非结构网格

2 问题分析

本文采用Jameson中心格式来求解二维Euler方程。在空间离散上采用的是有限体积法，时间上采用的是四步显式Runge-Kutta迭代求得最后的定常解。人工耗散项为守恒变量的二阶和四阶差分。边界条件采用的是无反射边界条件，并采用当地时间步长进行加速收敛。

2.1 基本方程

由于不考虑粘性，二维NS方程可简化为Euler方程。欧拉方程是质量，动量，能量守恒定理的表达。在边界为S，面积为 Ω 的二维区域，方程可以写成以下形式：

$$\frac{\partial}{\partial t} \iint_{\Omega} W d\Omega + \int_S (F dy - G dx) = 0 \quad (1)$$

其中， x, y 是笛卡尔坐标， W 是守恒量矢量， F, G 是流动矢量，具体形式如下：

$$W = \begin{bmatrix} \rho & \rho U & \rho V & \rho E \end{bmatrix}^T \quad (2)$$

$$F = \begin{bmatrix} \rho U & \rho U^2 + P & \rho UV & \rho UH \end{bmatrix}^T \quad (3)$$

$$G = \begin{bmatrix} \rho V & \rho UV & \rho V^2 + P & \rho VH \end{bmatrix}^T \quad (4)$$

ρ, P, H, E 分别是密度，压强，单位质量总焓和单位质量总能量； U, V 是速度矢量的笛卡尔坐标系下的分量。这些变量之间的关系如下：

$$\rho E = P/(\gamma - 1) + \rho(U^2 + V^2)/2 \quad (5)$$

$$\rho H = \rho E + P \quad (6)$$

2.2 空间离散

计算区域被划分为有限数量的非重叠单元，并且积分形式的守恒方程应用到每个单元。单元的一般形状如下图所示：

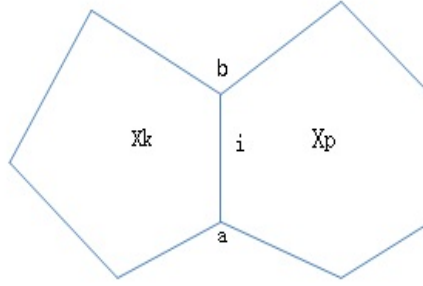


图 2: 离散的单元示意图

因为考虑到时间，任何单元的体积都可视为常数，所以(1)可写成如下形式：

$$\frac{\partial W}{\partial t} = - \frac{\int_S (F dy - G dx)}{\iint_{\Omega} d\Omega} \quad (7)$$

对于任一单元 k ，(7)可写成如下形式：

$$\frac{\partial W_k}{\partial t} = - \frac{Q_k}{\Omega_k} \quad (8)$$

其中， Ω_k, W_k, Q_k 分别是单元 k 的面积、守恒量矢量和(7)右边部分流动积分的离散近似值。

Ω_k 在网格输入文件中给出， Q_k 由以下近似给出：

$$Q_k = \sum_{i=1}^{edgeNum} (F dy - G dx)_i \quad (9)$$

dx, dy 为每边在坐标轴上的分量，按右手定则确定每条边的正方向， a, b 分别为每条边的起点和终点，计算规则如下：

$$dx_i = x_b - x_a \quad (10)$$

$$dy_i = y_b - y_a \quad (11)$$

为了计算出每条边上的流动矢量 F, G ，需要知道该边上的守恒量 W_i, W_i 由下式近似计算得到：

$$W_i = (W_k + W_p)/2 \quad (12)$$

(12)中的 W_k, W_p 分别为Fig2.2中的左右单元中心处的守恒矢量，从中也可见边的正方向的定义。

为了在计算过程中避免重复计算，我们引入如下辅助变量：

$$Z_i = U_i dy_i - V_i dx_i \quad (13)$$

对任一单元 k ，式(8)可写成如下形式：

$$\frac{\partial W_k}{\partial t} = - \frac{1}{\Omega_k} \sum_{i=1}^{edgeNum} \left[\rho Z \quad \rho U Z + P dy \quad \rho V Z - P dx \quad \rho H Z \right]^T \quad (14)$$

对于非结构网格，从边的角度入手来计算每个单元上的守恒矢量的变化量能避免重复计算，且对于网格形状来说具有通用性。

2.3 人工耗散

上述单元中心方案是非耗散的，所以任何误差（耗散误差，round-off误差）是没有被考虑进去的，并且振动可能在稳态解中出现。为了消除这些振动，人工耗散项被添加到式(8)的右边部分，对于单元k，式(8)变成如下形式：

$$\frac{\partial W_k}{\partial t} = -\frac{(Q_k - D_k)}{\Omega_k} \quad (15)$$

在现行工作中， D_k 被构建成保守变量 W_k 二阶微分和四阶微分的混和。 W_k 的四阶微分的部分被加到平滑的流动区域，但在有激波的区域不起作用，此时 W_k 的二阶微分项被接通用来抑制激波周围的振动，这项可以是相当大的。这种转换通过基于当地压强的二阶微分的激振感应器来获得。 D_k 的计算方法如下：

$$D_k = \sum_{i=1}^{edgeNum} d_i^{(2)} + \sum_{i=1}^{edgeNum} d_i^{(4)} \quad (16)$$

对于非结构网格，每条边上的保守变量的二阶和四阶微分由下式计算：

$$d_i^{(2)} = \alpha_i \varepsilon_i^{(2)} (W_p - W_k) \quad (17)$$

$$d_i^{(4)} = -\alpha_i \varepsilon_i^{(4)} (\nabla^2 W_p - \nabla^2 W_k) \quad (18)$$

在这里指标i是划分单元k和p的边， $\nabla^2 W_k$ 由下式计算得出：

$$\nabla^2 W_k = \sum_{j=1}^{edgeNum} (W_j - W_k) \quad (19)$$

这里 W_j 指的是与单元k相邻的单元上的守恒矢量，而不是单元k的边界上的守恒矢量。

在现今的工作中， $\varepsilon_i^{(2)}$ 和 $\varepsilon_i^{(4)}$ 由如下的一种简单方式构造得出：

$$\varepsilon_i^{(2)} = k^{(2)} \nu_i \quad (20)$$

$$\varepsilon_i^{(4)} = \max(0, k^{(4)} - \varepsilon_i^{(2)}) \quad (21)$$

这里 $k^{(2)}$ 和 $k^{(4)}$ 是两个由经验得出的常数，数值的典型范围是 $1/256 < k^{(2)} < 1/32$ 和 $1/2 < k^{(4)} < 10$ 。

ν_i 是激振感应器， α_i 是缩放比例因子，在现今的工作中，激振感应和缩放比例因子在边的基础上构造，只使用两个相邻单元k和p的流动变量，计算形式如下：

$$\nu_i = \frac{|P_p - P_k|}{|P_p + P_k|} \quad (22)$$

$$\alpha_i = |U dy - V dx| + c \sqrt{dx^2 + dy^2} \quad (23)$$

这里U，V和c分别是分界边上的速度与当地声速。

在边界附近（有效边界和外边界）对耗散项不正确的处理将导致解精度的严重损失。添加用来抑制振动的耗散项可以变得过于强，局部地，数值方案的顺序可被简化。从物理的观点来看，这种强耗散在边界附近产生了大量的伪数值熵。这引起了翼型表面的压强损失，而且积分载荷也会不正确。

2.4 时间离散

稳态解是通过对大系统的常微分方程的时间积分所得到的，可以写成如下形式：

$$\frac{dW_k}{dt} = -\frac{Q_k - D_k}{\Omega_k} \equiv R_k \quad (24)$$

其中 R_k 代表每个单元中心 k 的剩余误差（稳态解的偏移）。

方程(24)中的时间积分采用了四步Runge-Kutta方案。因为时间的精确度对于稳态解来说并不重要，这类方案只用于求解稳态的流场和稳定的阻尼特性。本文采用如下实施方案：

$$\begin{cases} W^{(0)} = W^n \\ W^{(m)} = W^{(0)} + \alpha_m \Delta t R^{(m-1)}, m = 1 \dots 4 \\ W^{n+1} = W^{(4)} \end{cases} \quad (25)$$

在这里 n 是当前时间步， $n+1$ 是下一个时间步。由于目前的计算机的计算能力相比于上世纪90年代有了长足的进步，而且计算耗散函数 $D^{(m)}$ 的时间复杂度是 $O(N)$ ，因此本文在迭代过程中的每一步都更新 $D^{(m)}$ ，最终 $R^{(m)}$ 由下式计算：

$$R^{(m)} = -\frac{Q^{(m)} - D^{(m)}}{\Omega} \quad (26)$$

方程组(25)中的系数 α_m 取值如下：

$$\alpha_1 = 1/4, \alpha_2 = 1/3, \alpha_3 = 1/2, \alpha_4 = 1$$

由于最终只要得到稳态解，为了加速计算，每个单元可以采用当地时间步长而非全局统一的时间步长。对于有任意形状单元的网格，单元 k 的当地时间步长可采用以下形式计算：

$$\Delta t_k = \frac{\Omega_k CFL}{\sum_{i=1}^{edgeNum} \alpha_i} \quad (27)$$

其中， α_i 是每条边上的缩放因子，由式(23)计算得出。

2.5 边界条件

对于物面，由于是无粘流动，所以边界上的速度只需满足“法向无穿透，切向滑移”即可。同时，壁面压强被等同于相邻的边界单元中心处压强。大量数值研究发现如果相邻于表面的单元足够小，并且如果人工耗散项被正确处理，这种对压强的弱条件的使用并不会对解的精度有很大影响。

对于远场边界，要求没有流出波再被反弹回计算区域。接下来是Jameson和Baker的方法，对于边界的法向流动，基于Riemann守恒，组成局域一维理论。本文采用的是Jameson的远场无反射边界条件（扰动波不反射回流场）。具体实施如下：

R^+, R^- 是两个Riemann不变量，取值如下：

$$R^+ = V_{ne} + \frac{2c_e}{\gamma - 1} \quad (28)$$

$$R^- = V_{n\infty} - \frac{2c_\infty}{\gamma - 1} \quad (29)$$

其中，下标 n 表示法向值， ∞ 表示来流值， e 表示由该边紧邻的单元上的流动变量外插得到的值（即近似认为边界上的值为相邻单元中对应的物理量的值），边界上的法向速度 V_n 和音速 c 可以由这两个Riemann不变量求得：

$$V_n = \frac{1}{2}(R^+ + R^-) \quad (30)$$

$$c = \frac{\gamma - 1}{4}(R^+ - R^-) \quad (31)$$

可以将远场边界条件分为如下4种情况:

(1) 亚音速入流($M_{n\infty} < 1, V_{ne} < 0$)

$$\begin{aligned} V_n &= \frac{1}{2}(R^+ + R^-) \\ V_t &= V_{t\infty} \\ c &= \frac{\gamma - 1}{4}(R^+ - R^-) \\ s &= s_\infty \end{aligned} \quad (32)$$

(2) 亚音速出流($M_{n\infty} < 1, V_{ne} > 0$)

$$\begin{aligned} V_n &= \frac{1}{2}(R^+ + R^-) \\ V_t &= V_{te} \\ c &= \frac{\gamma - 1}{4}(R^+ - R^-) \\ s &= s_e \end{aligned} \quad (33)$$

(3) 超音速入流($M_{n\infty} > 1, V_{ne} < 0$)

$$\begin{aligned} \rho &= \rho_\infty \\ u &= u_\infty \\ v &= v_\infty \\ p &= p_\infty \end{aligned} \quad (34)$$

(4) 超音速出流($M_{n\infty} > 1, V_{ne} > 0$)

$$\begin{aligned} \rho &= \rho_e \\ u &= u_e \\ v &= v_e \\ p &= p_e \end{aligned} \quad (35)$$

根据上述值可以计算出边界上的各个物理量的值, 进一步可以计算出远场边界上守恒量的值。

3 编程实现

3.1 网格文件说明

给定的计算网格是以三角形为基本单元的非结构网格, 网格文件第一行是3个整数, 分别表示节点的个数、边的个数和单元的个数。接下来依次给出每个点的x,y坐标。

紧接着, 输入文件依次给出每条边的起点序号、终点序号、该边左边单元的序号和该边右边单元的序号, 输入文件保证左单元的序号一定是正的, 也即正方向由右手定则确定, 右单元的序号可能是负的, 若为-1则表示该边是物面边, 若为-2则表示该边是远场边。由于输入文件中的序号都是以1为起始序号的, 为了计算方便, 本文在程序中统一将其转换为以0为起始序号。

最后, 输入文件依次给出每个单元的信息, 每行给出该单元3个顶点的序号, 这些顶点序号仍然是以1为起始的, 同样, 我们将其转换为以0为起始的。

3.2 数据存储结构

显然，节点、边和单元是程序的基本组成部分。本程序使用C++实现，采用面向对象的方法为节点、边和单元建模，将其分别处理成一个类，在类中封装了相应的属性信息和方法。具体实现如下：

节点的属性较为简单，只需存储坐标即可。

```

1 class Point // 只列出与流动计算相关部分，未显示出构造函数等信息
2 {
3 public:
4     vector<double> coordinate;
5
6     double DistanceTo(const Point &rhs) const; // 到另一点的几何距离
7     vector<double> DeltaTo(const Point &rhs) const; // 到另一点的矢量
8 };

```

单元上的信息相对边要少一些，除了基本的顶点信息和单元中心处的流动变量，还存放了一定辅助量用于计算残差和Runge-Kutta迭代，主要部分如下：

```

1 class Cell // 只列出与流动计算相关部分，未显示出构造函数等信息
2 {
3 public:
4     vector<int> point; // 顶点序号集
5     double volum; // 单元面积
6     vector<double> w; // rho, rho*u, rho*v, rho*E
7     vector<double> w_next;
8     vector<double> physicalVar; // rho, u, v, p
9     vector<double> pre_physicalVar;
10    double localTimeStep;
11    vector<double> Q; // 单元的对流通量
12    vector<double> D; // 单元的耗散通量
13    vector<double> ddw; // 守恒量的Laplace
14
15    void updatePrevRecord();
16    double getDensity() const;
17    double getDensityDiff() const;
18    double getVelocity() const;
19    double getVelocityDiff() const;
20    double getPressure() const;
21    double getPressureDiff() const;
22 };

```

边的属性较为复杂，不仅包含了构成边的基本信息，还要存储很多流场中的物理量的信息和人工耗散项的信息，同时为了加速计算，采用了一定程度的“空间换时间”策略，保存了每条边的长度、向量表示、单位外法向量等用于加速计算的辅助信息。


```

1  class Edge//只列出与流动计算相关部分，未显示出构造函数等信息
2  {
3  private:
4      void calcFarBound(); //根据无反射边界条件计算远场边得到,w_av, rho, u, v, p, c
5  public:
6      int start, end;
7      int leftCell, rightCell;
8      double len;
9      vector<double> delta;
10     vector<double> delta_n; //单位外法向量
11     vector<double> w_av; //两边单元的物理量的平均
12     vector<double> physicalVar; //边上的物理量: rho, u, v, p
13     double c; //当地声速
14     double Z; //用于辅助计算
15     vector<double> convective_flux; //该边上的对流通量
16     vector<double> w_diff; //两边单元物理量之差，没有右则不算
17     double ScalingFactor;
18     double v; //激波探测器
19     double eps2, eps4; //流场探测器
20     vector<double> d2, d4; //components of the dissipation_flux
21
22     Point getMidPoint() const; //取得中点
23     void calcBasicVariables(); //计算该边上的，根据不同类型做不同处理w_av
24     void calcConvection(); //计算该边上的对流通量Q
25     void addConvectionToAdjcentCell(); //将该边上的对流通量累加到相关联的单元
26     void calc_d4(); //计算d4
27     void addDissipationToAdjcentCell(); //将该边上的耗散通量累加到相关联的单元
28 };

```

3.3 程序流程

程序主要分3步：首先读入网格并初始化流场变量，然后按设定的最大迭代次数做Runge-Kutta迭代进行时间推进，最后到达指定迭代步数或残差收敛到一定量级后输出结果。

初始化的过程中提示用户输入来流数据，如马赫数、攻角、静压和密度等物理量，然后构造格式化的输出文件并读入网格信息。

输出阶段则将最终流场结果按照Tecplot的数据格式要求进行输出，同时计算翼型上下表面的压力分布，并输出到相应的文件中。

时间推进过程是本算法的核心所在。在每个时间步中先进行预处理，然后做Runge-Kutta迭代，最后输出当前时间步计算得到的结果。这里面Runge-Kutta迭代部分是关键。Runge-Kutta迭代部分又由4个子过程组成，这4个子过程具有相似性，因此本文着重介绍单个Runge-Kutta子迭代步的流程，该流程如下图所示：

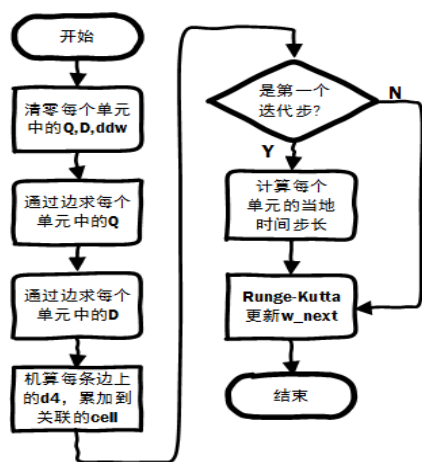
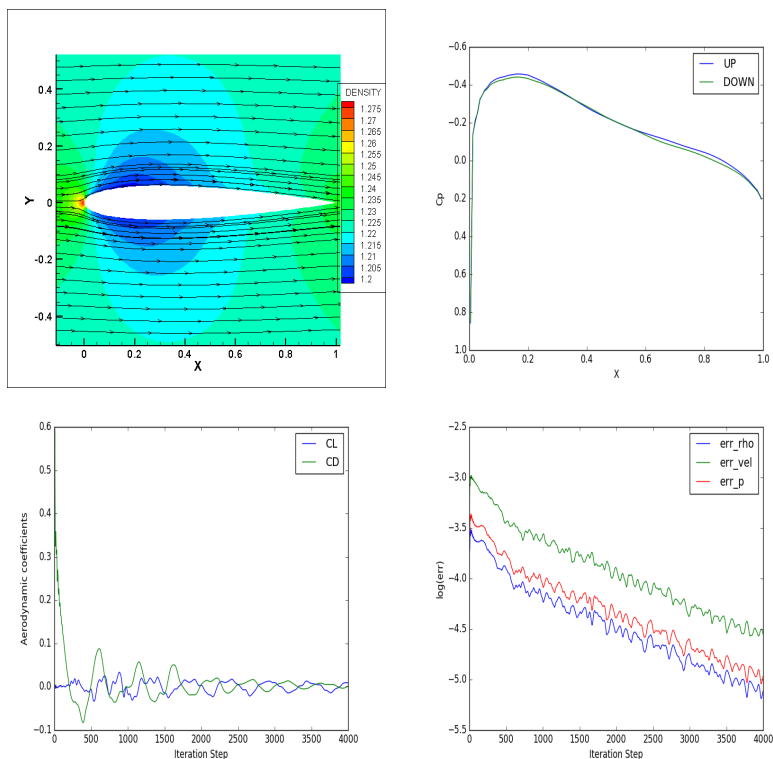


图 3: Runge-Kutta子迭代步流程

4 结果分析

4.1 $Ma=0.3$

来流迎角为 0° 时得到的结果如下图所示。

图 4: $Ma = 0.3, \alpha = 0^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

由于来流和翼型都是对称的，因此上下流场也对称，翼型上不存在压差，故没有升力。

来流迎角为 3° 时得到的结果如下图所示。从图中可见当来流有迎角时，翼型上下表面产生压差。虽然计算迭代的过程中存在波动，但总体上趋于收敛。

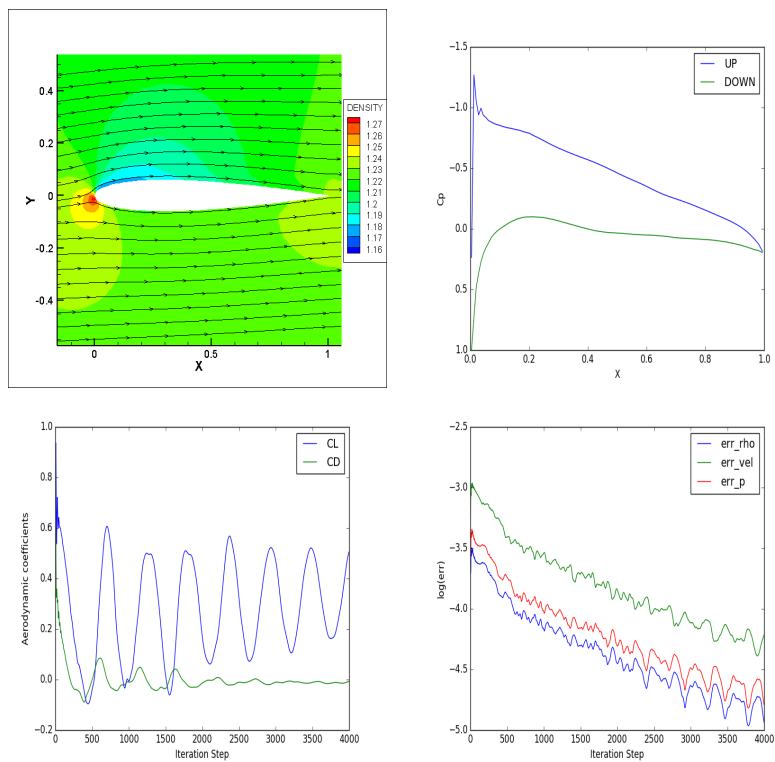


图 5: $Ma = 0.3, \alpha = 3^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

来流迎角为 6° 时得到的结果如下图所示。

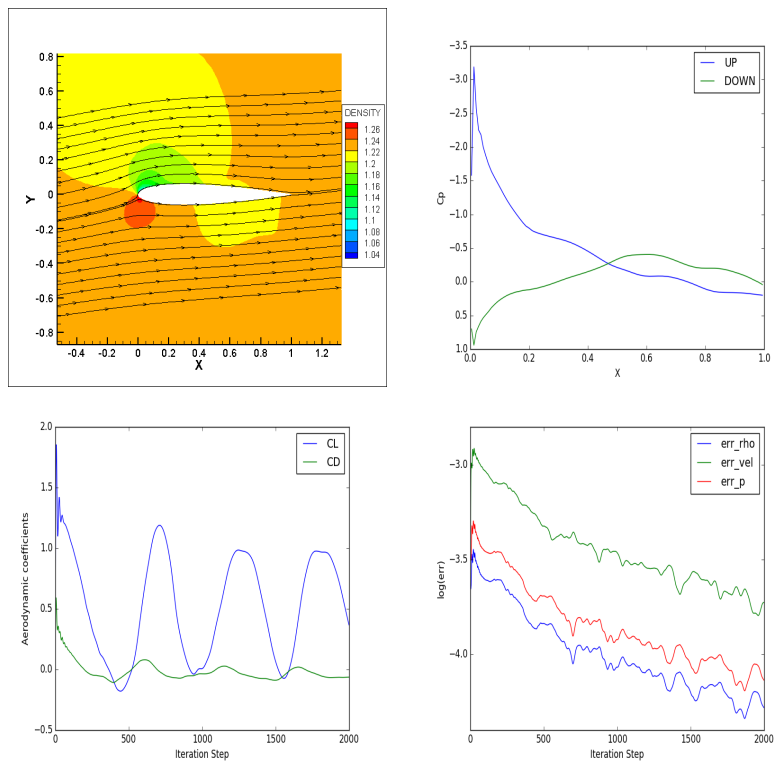


图 6: $Ma = 0.3, \alpha = 6^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

可见当来流迎角进一步增大时, CL 也随之增大。 $k^{(2)}, k^{(4)}$ 的取值偏小, 导致迭代过程中存在明显波动。

来流迎角为 9° 时得到的结果如下所示。可见升力系数随着迎角的增大而增加，符合已知实验结果。

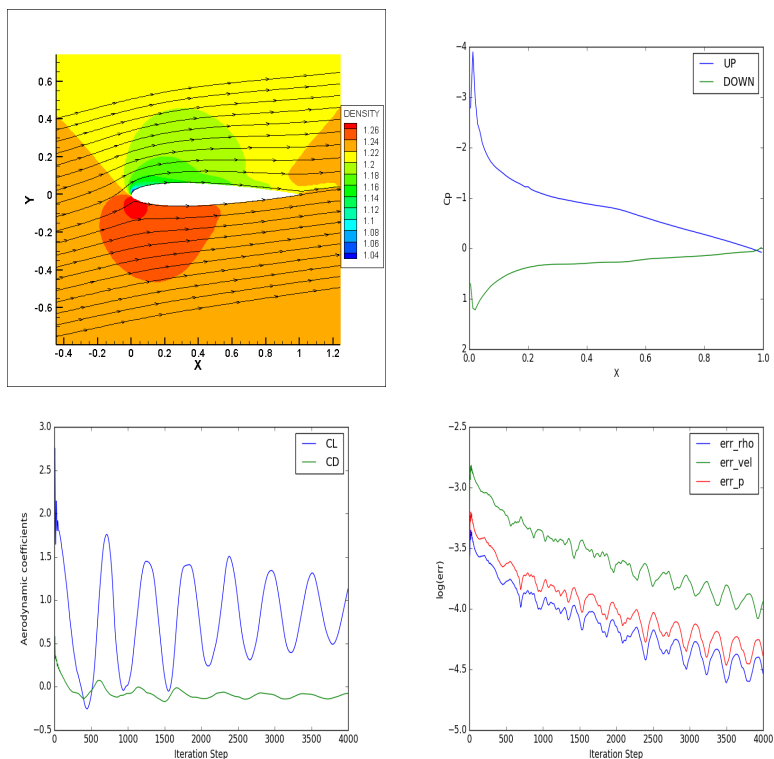


图 7: $Ma = 0.3, \alpha = 9^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

4.2 $Ma=0.8$

来流迎角为 0° 时得到的结果如下图所示。由于来流和翼型都是对称的，因此上下流场也对称，翼型上不存在压差，故没有升力。但计算过程中有较大波动，体现了跨音速附近的不稳定性。

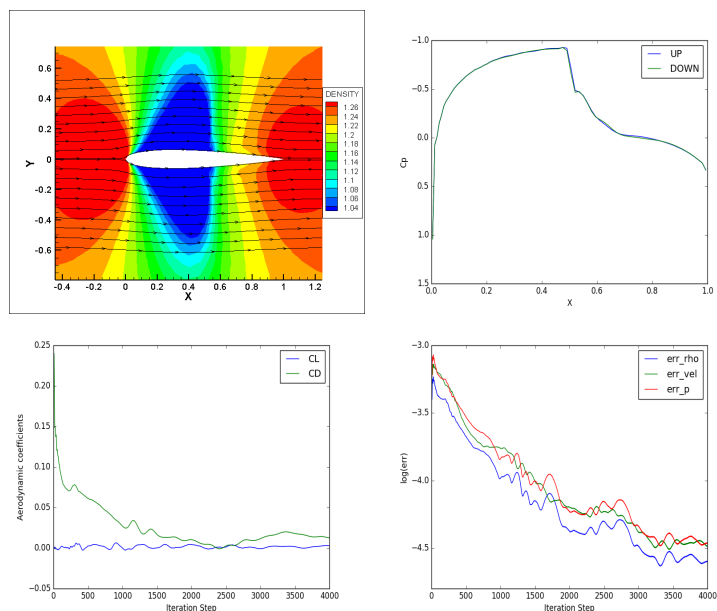


图 8: $Ma = 0.8, \alpha = 0^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

来流迎角为 3° 时得到的结果如下图所示。从压力分布图上可以看出在上表面中部存在一道激波。

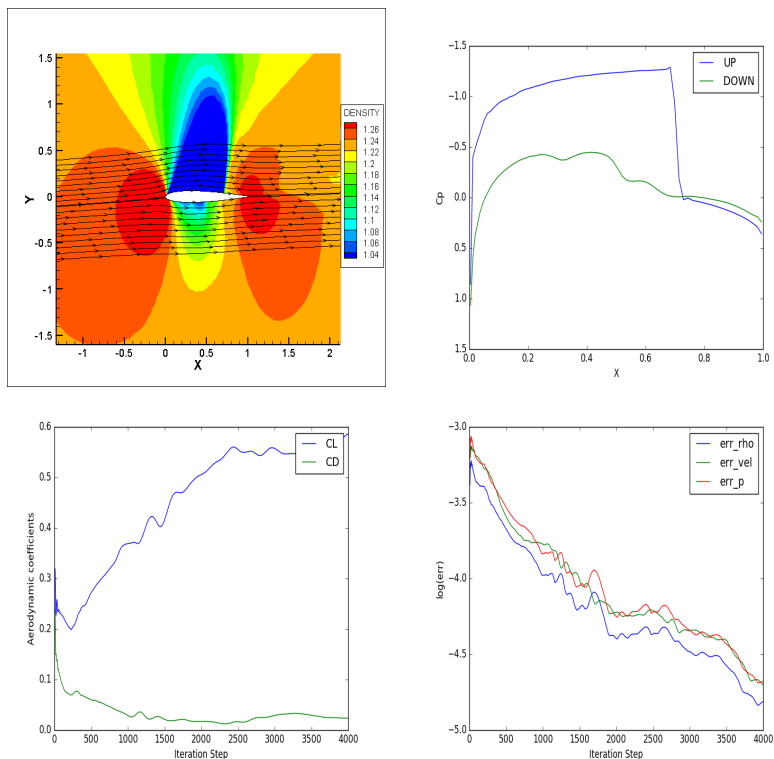


图 9: $Ma = 0.8, \alpha = 3^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

来流迎角为 6° 时得到的结果如下图所示。随着迎角的增大，激波也随之增强。

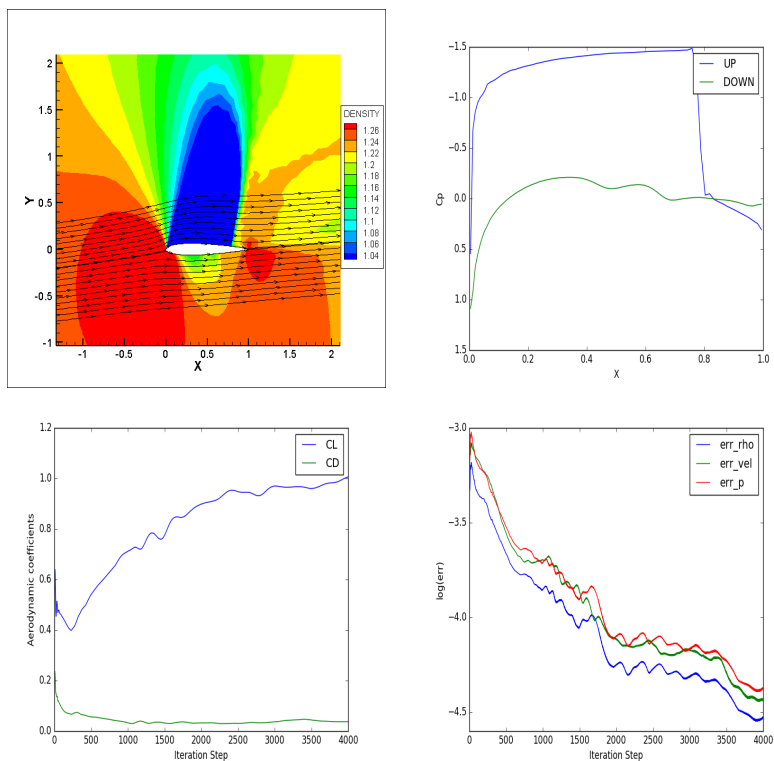


图 10: $Ma = 0.8, \alpha = 6^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

来流迎角为 9° 时得到的结果如下图所示。流动规律与迎角为 6° 时相似。

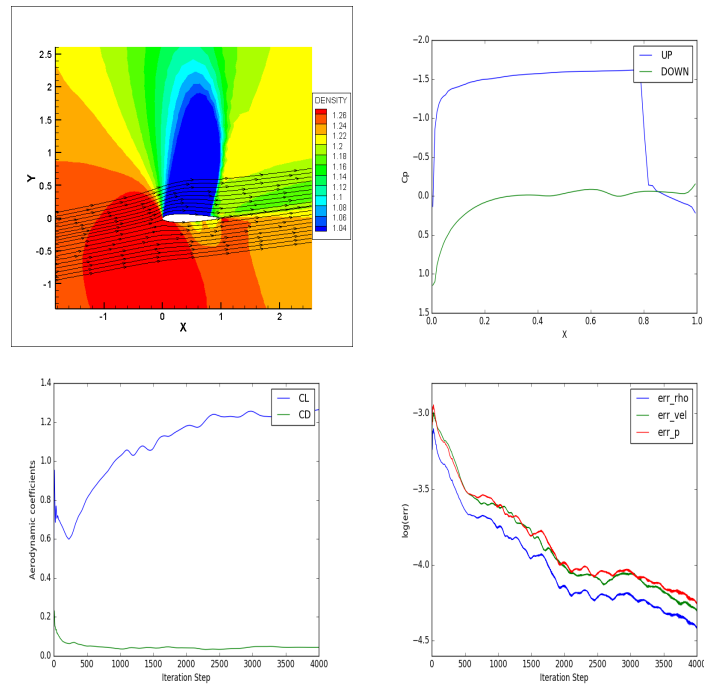


图 11: $Ma = 0.8, \alpha = 9^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

4.3 $Ma=1.2$

来流迎角为 0° 时得到的结果如下图所示。由于来流和翼型都是对称的,因此上下流场也对称,翼型上不存在压差,故没有升力,但由于存在激波,阻力不为0。

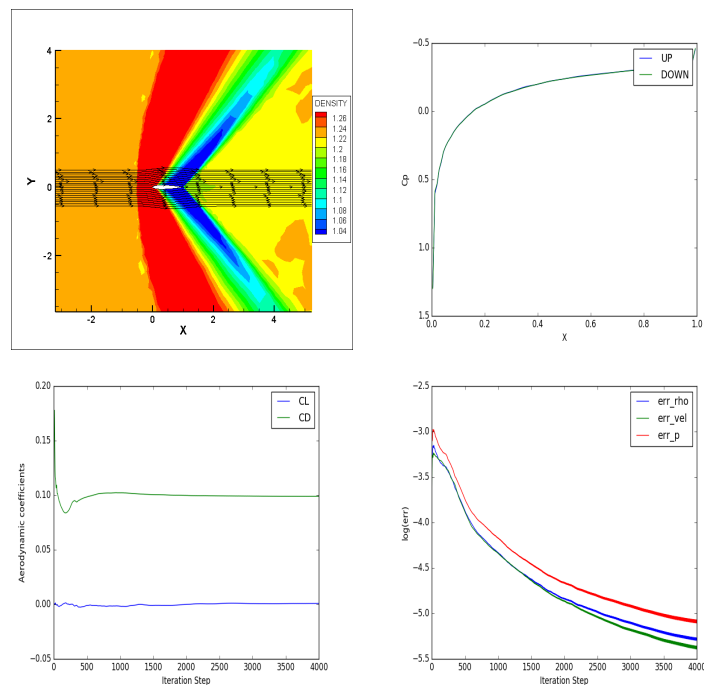


图 12: $Ma = 1.2, \alpha = 0^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

来流迎角为 3° 时得到的结果如下图所示。由于存在迎角和激波，升力和阻力均不为0。计算过程较为平滑。

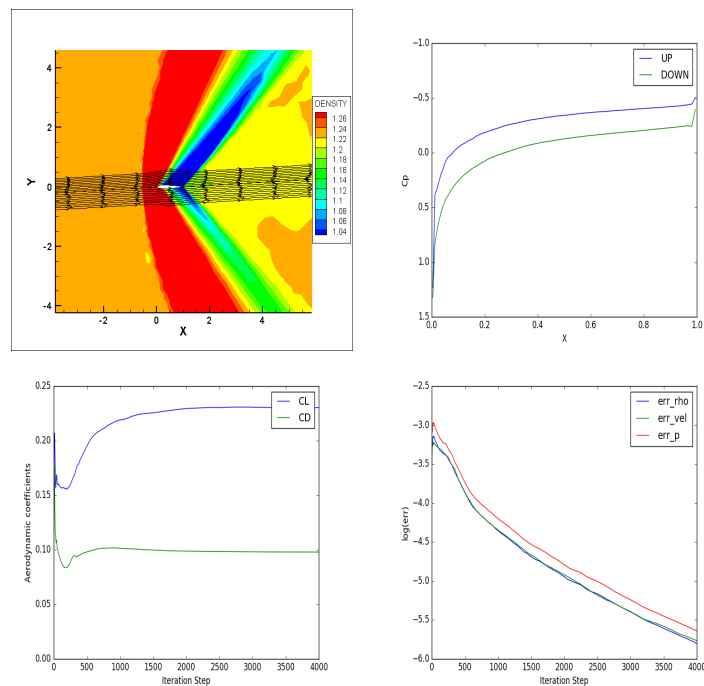


图 13: $Ma = 1.2, \alpha = 3^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

来流迎角为 6° 时得到的结果如下所示。计算得出的升力系数与经验公式的结果相吻合。

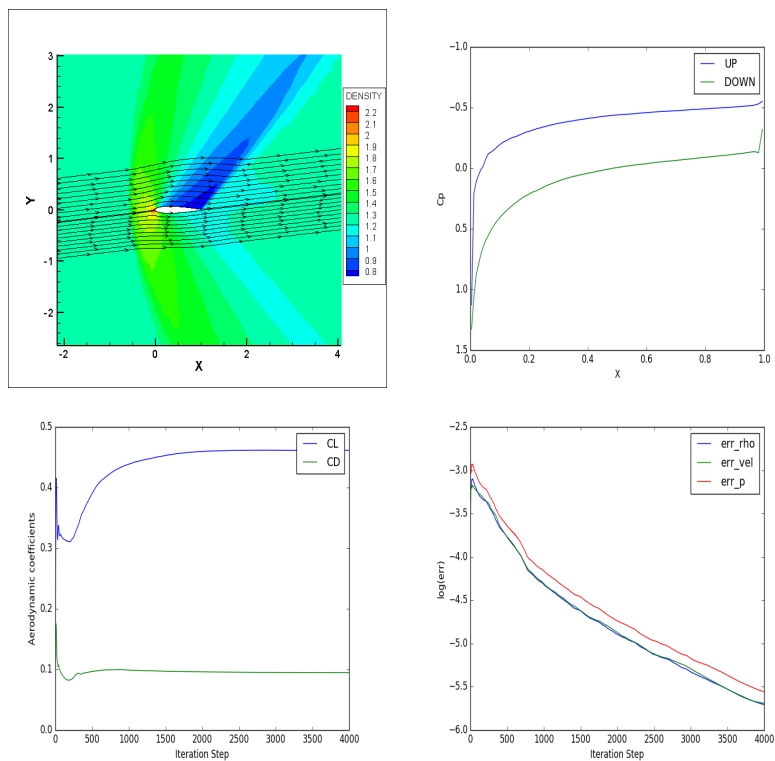


图 14: $Ma = 1.2, \alpha = 6^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

来流迎角为 9° 时得到的结果如下图所示。可见在超音速时计算结果也符合实际。

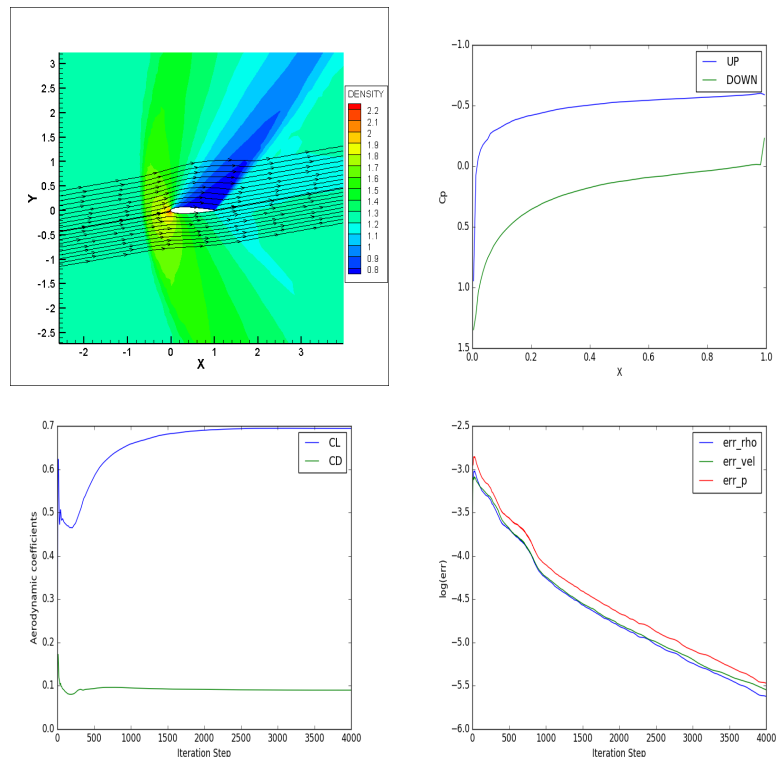


图 15: $Ma = 1.2, \alpha = 9^\circ$ 翼型周围的流场、翼型压力分布、气动力系数与残差

5 总结

本文使用了经典的有限体积方法，基于Jameson中心格式求解了二维无粘流动的Euler方程。给出了在低速、跨音速、超音速三种情况下的计算仿真结果。结果表明在加入人工粘性项后，数值计算可以有效收敛，并得到与实验相符合的结果。但是这一结果的取得与给定的人工粘性项系数有较大关系，若人工耗散项较小则会出现震荡现象，而人工耗散系数的选择还是依靠经验的，这也是CFD计算的局限性。Jameson格式在处理空间离散上比较简便，编程实现也比较容易，是一种比较可行的方法。