

OpenCFD-EC 理论手册

(2015-3 更新)

李新亮

中国科学院力学研究所

版权声明

OpenCFD-EC (Open Computational Fluid Dynamic code for Engineering Computation) 是可压缩 Navier-Stokes 方程多块结构网格有限体积求解器。主要用于计算工程问题。该软件是作者 OpenCFD 计划的一部分。

作者的 OpenCFD 计划包含本软件(OpenCFD-EC)及另外一个软件 OpenCFD-SC (Open Computational Fluid Dynamic code for Scientific Computation, 即原先的 Hoam-OpenCFD)。OpenCFD-SC 是可压缩 N-S 方程高精度差分求解器, 主要用来计算科学问题。

OpenCFD-EC 及 OpenCFD-SC 是由作者李新亮开发的计算流体力学 (CFD) 软件。目前该软件尚未开源, 未获作者许可, 获得该软件代码的用户请不要泄露给第三方。

如在科研或工程计算中使用 OpenCFD-EC 及 OpenCFD-SC, 请务必在论文及报告中进行标注。即在文中致谢并引用作者的论文 (作者的论文可使用 SCI 搜索作者=“LIXL AND FUDX” 即可)。

参考文献:

J. Blazek: Computational Fluid Dynamics: Principles and Applications, Elsevier 2005

傅德薰 主编 《计算空气动力学》

任玉新等 《计算流体力学基础》

李新亮 《计算流体力学》 课件

(下载地址: <http://pan.baidu.com/s/1kTxHAjP>)

讨论版: www.cfluid.com (流体中文网) -> “流体论坛” -> “CFD 基础理”)

朱自强等: 《应用计算流体力学》

E. F. Toro: Riemann Solvers and numerical methods for fluid dynamics

G. KARYPIS, V. KUMAR, *A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs*, *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, pp. 359—392, 1999

《CFL3D Version 5.0 manual》 <http://cfl3d.larc.nasa.gov/> (Version 5.0 manual, TurbEq.pdf)

关于本手册及相关软件的更新请关注作者网盘:

<http://pan.baidu.com/s/1kTxHAjP>

目 录

第一章 二维求解器	4
1.1 基本方程	4
1.2 无粘项的离散	6
1.3 粘性项的离散	11
1.4 网格的多块对接	14
第二章 三维求解器	15
2.1 基本方程	15
2.2 无粘项的离散	16
2.3 粘性项的计算	19
2.4 块的边界条件及连接信息	22
第三章 时间推进及加速收敛技术	28
3.1 显式时间推进格式	
3.2 局部时间步长法	
3.3 LU-SGS 隐格式	
3.4 双时间步长法	
3.5 多重网格法	
第四章 湍流模型	38
4.1 BL 模型	
4.2 SA 模型	
4.3 SST k- ω 模型	
第五章 有限差分-有限体积混合方法	50
5.1 有限差分-有限体积混合方法的基本原理	
5.2 有限差分模块基本原理	
第六章 OpenCFD-EC 的数据结构及流程	55
6.1 二维程序	
6.2 三维程序	
第七章 并行计算	60
7.1 OpenMP 并行	
7.2 MPI 并行	
第八章 文件格式及转换	
8.1 Gridgen 的 Generic 格式网格文件	
8.2 CGNS 格式网格文件	

第九章 后处理

第十章 算例

附录 1: 叶轮机械内流的计算

第一章 二维求解器

§1.1 基本方程

本章以二维问题为例，说明本软件求解器的理论方法。

控制方程为二维守恒型 Navier-Stokes 方程

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_1(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{F}_2(\mathbf{U})}{\partial y} = \frac{\partial \mathbf{F}_{v1}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{F}_{v2}(\mathbf{U})}{\partial y} \quad (1.1.1)$$

本软件采用无量纲形式计算。密度、速度、温度和压力分别采用特征密度 $\tilde{\rho}_\infty$ ，特征速度 \tilde{U}_∞ ，特征温度 \tilde{T}_∞ 及动压 $\tilde{\rho}_\infty \tilde{U}_\infty^2$ 无量纲化。

无量纲 N-S 方程的具体形式见傅德薰《计算空气动力学》。

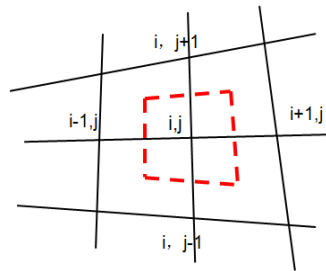
在以(I,J)点为中心的控制体上，对 N-S 方程进行积分，得到如下方程

$$\frac{\partial \bar{\mathbf{U}}_{IJ}}{\partial t} + \frac{1}{\Omega_{IJ}} \oint_{\partial \Omega} \mathbf{F} \cdot \mathbf{n} ds - \frac{1}{\Omega_{IJ}} \oint_{\partial \Omega} \mathbf{F}_v \cdot \mathbf{n} ds = 0 \quad (1.1.2)$$

左端第 1 项为时间导数项，第 2 项为无粘项，第 3 项为粘性项。

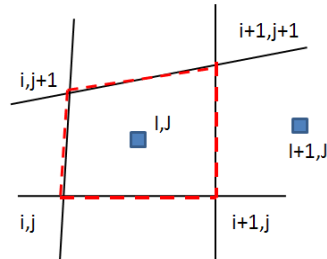
控制体的选取主要有网格中心型及节点中心型两种方式（见图 1）。相对而言，网格中心型控制体在处理角点、壁面及交接面时具有一定的优势（见图 2），因而**本程序采用网格中心型控制体**。

在网格中心型控制体中，物理量（密度、速度、压力等）存储在控制体中心。本手册约定：**下标 (i,j) 表示网格节点，下标 (I,J) 表示控制体中心**。



节点中心型控制体

物理量储存在节点 (Vertex) 上



网格中心型控制体

物理量储存在网格中心上

图 1.1 网格及控制体示意图

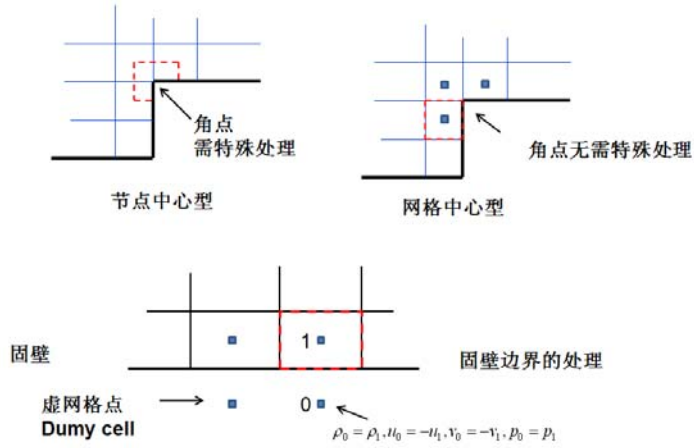


图 1.2 网格中心型及节点中心型控制体积的对比

其中，**控制体中心**定义为：

$$\vec{r}_c = \frac{1}{\Omega} \int_{\Omega} \vec{r} d\Omega$$

即该四边形的**重心**。

具体计算时，四边形 ABCD 的中心可以用三角形 ABD 及 BCD 的中心加权平均得到。即：

$$\vec{r}_c = \frac{\Omega_{ABD} \vec{r}_{c,ABD} + \Omega_{BCD} \vec{r}_{c,BCD}}{\Omega_{ABD} + \Omega_{BCD}}$$

其中，三角形中心的坐标为三个顶点坐标的代数平均。即：

$$\vec{r}_{c,ABD} = \frac{1}{3}(\vec{r}_A + \vec{r}_B + \vec{r}_D)$$

三角形的面积公式为：

$$\Omega_{ABD} = \frac{1}{2} |\vec{r}_{AB} \times \vec{r}_{AD}| = \frac{1}{2} [(x_B - x_A)(y_D - y_A) - (y_B - y_A)(x_D - x_A)]$$

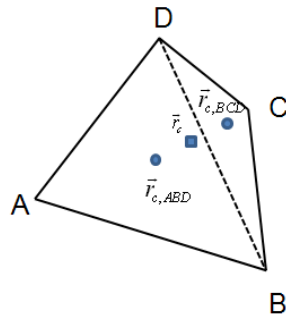


图 1.3 示意图： 四边形 ABCD 的中心

§1.2 无粘项的离散

1.2.1 积分方程

(1.1.2) 式左侧第 2 项为无粘项，其表达式如下：

$$\frac{1}{\Omega_U} \oint_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} ds = \frac{1}{\Omega_U} (\mathbf{H}_{I+1/2,J} - \mathbf{H}_{I-1/2,J} + \mathbf{H}_{I,J+1/2} - \mathbf{H}_{I,J-1/2}) \quad (1.2.1)$$

其中 Ω_U 是以包围中心点 (I,J) 的控制体 (图 1 中红色虚线围成四边形) 的面积。具体计算时，可用该四边形对角线的向量进行叉乘获得。

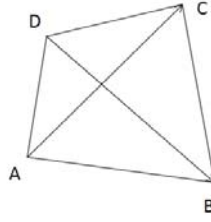


图 1.4 示意图：四边形的面积 $= \frac{1}{2} |\overline{AC} \times \overline{DB}|$

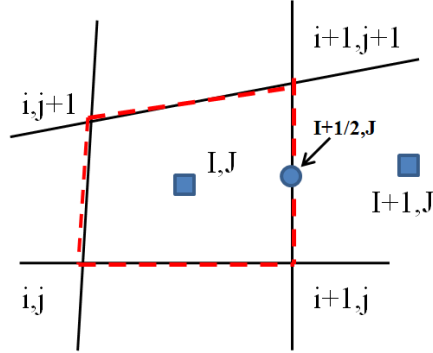


图 1.5 控制体示意图

穿过控制体右边界面 (图 1.5 中箭头所指的界面) 的无粘通量的表达式为

$$H_{I+1/2,J} = [n_1 F_1 + n_2 F_2]_{I+1/2,J} \Delta s_{I+1/2,J} \quad (1.2.2)$$

其中：

$$\mathbf{F}_1 = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E + p)u \end{bmatrix} \quad \mathbf{F}_2 = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E + p)v \end{bmatrix} \quad (1.2.3)$$

$\mathbf{n} = (n_1, n_2)^T$ 为控制体右边界面的法方向， $\Delta s_{I+1/2,J}$ 为该界面的长度 (在三维情况下为面积)。

控制体其他边界面通量的计算方法与右边界的计算方法相同。

无粘通量计算的关键是计算 $(I+1/2, J)$ 点处的通量 $F_{I+1/2, J}$ 。如采用中心型格式，该值可左右、右两侧值平均获得，即：

$$F_{I+1/2, J} = (F_{I, J} + F_{I+1, J}) / 2 \quad (1.2.4)$$

或

$$F_{I+1/2, J} = F(U_{I, J} + U_{I+1, J}) / 2 \quad (1.2.5)$$

直接使用(1.2.4)或(1.2.5)会带来不稳定。为了消除数值振荡，通常在(1.2.4)或(1.2.5)式右端加入人工粘性项，以便于稳定程序。

即使添加了人工粘性，采用中心格式计算的 Mach 数仍不会太高（例如不超过 3）。**本程序采用迎风格式计算无粘项。**迎风格式需要把无粘通量按照特征方向进行分解（以便于确定“迎风”方向）。根据分解形式不同，可分为流通矢量分裂(FVS)及通量差分分裂(FDS)两类。

常用的流通矢量分裂(FVS)包括 Lax-Friedrichs 分裂，Steger-Warming 分裂，Van Leer 分裂及 Liou-Steffen 分裂等。Lax-Friedrichs 分裂最为简单，且能保证函数的光滑性，但其耗散最大。Steger-Warming 分裂耗散较小，但无法保证函数音速点的光滑性。Van Leer 分裂克服了 Steger-Warming 分裂的缺点，但其耗散略大。Liou-Steffen 分裂即 AUSM 方法，是在 Van Leer 分裂基础上，将压力项单独处理的一种方法，在目前得到了广泛的应用。

通量差分分裂(FDS)通过 Riemann 解来计算数值通量，因为更好地利用了双曲方程的特征方向，因而其激波捕捉能力更强，数值振荡更小。但其计算量要大于流通矢量分裂。常用的 FDS 方法包括精确 Riemann 解(Godunov 方法)，Roe 近似 Riemann 解及 HLL/HLLC 近似 Riemann 解等。

本软件提供 Steger-Warming FVS，HLL/HLLC 及 Roe FDS 四种通量分裂方法。

1.2.2 通量差分分裂(FDS)方法

先重构出控制界面 $(I+1/2, J)$ 上的值。在迎风型方法中，利用（偏）左侧及（偏）右侧网格点，重构出该点的左值 $U_{i+1/2}^L$ 及右值 $U_{i+1/2}^R$ 。可利用原始变量、守恒变量或特征变量进行重构。采用特征变量进行重构数值振荡最小，但计算量最大。本软件采用**守恒变量**进行重构。

重构方法与有限差分法构造差分格式的形式完全相同。本软件采用 2 阶或 3 阶 3 阶迎风偏斜重构（见 2.3 节）。

重构出 $U_{I+1/2}^L$ 及 $U_{I+1/2}^R$ 以后，求解 Riemann 问题，即一维 Euler 方程及如下初值条件：

$$U = \begin{cases} U_{I+1/2}^L & \text{if } x < x_{I+1/2} \\ U_{I+1/2}^R & \text{otherwise} \end{cases}$$

即可得到穿过 $(i+1/2, j)$ 界面的通量 $H_{i+1/2, j}$ 。

需要注意，由于本问题属于二维问题，可通过坐标旋转，转化成为垂直界面方向的（扩展）一维 Riemann 问题，然后求解。具体公式见 任玉新 《计算流体力学基础》

(p164-168).

本软件目前使用 HLL/HLLC 积分型近似 Riemann 求解器及 Roe 微分型近似 Riemann 求解器计算通量。具体公式见 Toro: Riemann Solvers and numerical methods for fluid dynamics。

1.2.3 流通矢量分裂(FVS)方法

与 FDS 方法相同，首先重构出界面 $(I+1/2, J)$ 处的函数值： $U_{I+1/2}^L$ 及 $U_{I+1/2}^R$ 。

利用 Steger-Warming 流通矢量分裂，将控制界面上的通量分解为正通量及负通量：

$$H_{I+1/2, J} = H_{I+1/2, J}^+ + H_{I+1/2, J}^-$$

具体公式见 傅德薰《计算空气动力学》158-159 页（请留意书中的印刷错误）。

(2.5) 式中的系数为：

其中正通量用 $U_{I+1/2}^L$ 计算出，负通量用 $U_{I+1/2}^R$ 计算出，即：

$$H_{I+1/2, J} = H_{I+1/2, J}^+(U_{I+1/2, J}^L) + H_{I+1/2, J}^-(U_{I+1/2, J}^R)$$

与 FDS 方法相同，对于二维问题，需要进行坐标旋转，转化为“扩展一维问题”。

1.2.4 重构

重构是利用节点上的控制体平均值 U_i 计算出控制体界面（半点）上的值 $U_{i+1/2}$ 。应当注意重构与插值的区别（见作者的《计算流体力学》课件）。对于不超过 2 阶精度的重构，由于节点值与控制体平均值相同，因此重构形式与插值相同。但对于高阶方法二者不同。当网格充分光滑时，有限体积法与有限差分法重构形式相同，有限差分法中的公式可直接应用。

本软件使用的重构方法有如下四种：

1) 2 阶 NND 格式 (NND2)

$$\begin{aligned} U_{I+1/2}^L &= U_I + 1/2 \min \text{mod}(U_{I+1} - U_I, U_I - U_{I-1}) \\ U_{I+1/2}^R &= U_{I+1} - 1/2 \min \text{mod}(U_{I+1} - U_I, U_{I+2} - U_{I+1}) \end{aligned} \quad (1.2.6)$$

2) 3 阶迎风偏斜格式(UCD3)

$$\begin{aligned} U_{I+1/2}^L &= (-U_{I-1} + 5U_I + 2U_{I+1}) / 6 \\ U_{I+1/2}^R &= (2U_I + 5U_{I+1} - U_{I+2}) / 6 \end{aligned} \quad (1.2.7)$$

3) 3 阶 WENO 格式 (WENO3)

$$\begin{aligned}
IS_1 &= (U_I - U_{I-1})^2, IS_2 = (U_{I+1} - U_I)^2 \\
\alpha_k &= C_k / (IS_k + \varepsilon)^2, \omega_k = \alpha_k / (\alpha_1 + \alpha_2) \\
C_1 &= 1/3, C_2 = 2/3 \\
U_{I+1/2}^L &= \omega_1(-U_{I-1}/2 + 3/2 U_I) + \omega_2(U_I/2 + U_{I+1}/2)
\end{aligned} \tag{1.2.8}$$

4) 3 阶 MUSCL 格式 (MUSCL3) (使用 Van Albada 限制器)

$$U_{I+1/2}^L = U_I + s/4[(1-s/3)(U_I - U_{I-1}) + (1+s/3)(U_{I+1} - U_I)] \tag{1.2.9}$$

S 为 Van Albada 限制器, 具体形式为:

$$s = \frac{2(U_I - U_{I-1})(U_{I+1} - U_I) + \varepsilon}{(U_I - U_{I-1})^2 + (U_{I+1} - U_I)^2 + \varepsilon}$$

$\varepsilon = 10^{-6}$ 为一小量, 防止上式分母为 0.

容易看出, 在光滑区, $U_2 - U_1$ 与 $U_3 - U_2$ 相差不大, s 趋近于 1, 这时 3 阶 MUSCL 格式趋近于 3 阶普通迎风差分格式。

在剧烈震荡区, $U_2 - U_1$ 与 $U_3 - U_2$ 相差很大, s 趋近于 0, 这时 3 阶 MUSCL 格式趋近于 1 阶迎风格式。

同理, $I+1/2$ 出函数的右值为:

$$\begin{aligned}
U_{I+1/2}^R &= U_{I+1} - s/4[(1-s/3)(U_{I+2} - U_{I+1}) + (1+s/3)(U_{I+1} - U_I)] \\
s &= \frac{2(U_{I+1} - U_I)(U_{I+2} - U_{I+1}) + \varepsilon}{(U_{I+1} - U_I)^2 + (U_{I+2} - U_{I+1})^2 + \varepsilon}
\end{aligned}$$

1.2.5 原始变量、守恒变量与特征变量的重构

本软件支持使用原始变量、守恒变量或特征变量来重构出半点 $I+1/2$ 处的值。

所谓原始变量, 指密度、速度及压力。本软件中为网格内的平均密度、平均速度及平均压力。在二阶精度范围内, 可认为是网格中心处 (I 点) 的值。因为这些量可通过实验直接观测, 是最直观的变量, 因此称为“原始变量”。原始变量为:

$$\mathbf{q}_I = \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix}_I$$

所谓守恒变量，指的是密度（质量密度）、动量密度及能量密度。本软件仍然使用的是网格中心的值（更严格的讲上是网格内的平均值）。因为这些量满足质量守恒、动量守恒及能量守恒关系式（1.1），因此成为“守恒变量”。守恒变量为：

$$\mathbf{U}_I = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}_I$$

其中能量密度的表达式为：

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

为单位体积内的总能（内能+动能）。

特征变量是根据双曲方程特征理论，通过对守恒变量进行线性变换得到的一组新变量。即：

$$\mathbf{V}_I = \mathbf{S}_{I+1/2} \mathbf{U}_I \quad (1.2.10)$$

其中矩阵 \mathbf{S} 为 Jacobi 矩阵 $\mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}}$ 的左特征矩阵（ $\mathbf{A} = \mathbf{S}^{-1} \mathbf{\Lambda} \mathbf{S}$ ）。其具体表达式见傅德薰《计算空气动力学》第 4.7 节（158-162 页，需留意其中的印刷错误，公式以本软件为准）。

对于常系数偏微方程组，根据特性理论，采用特征变量可以把方程组解耦，变换成若干独立的单变量微分方程求解。对于像 $\mathbf{N-S}$ 方程这样复杂的偏微分方程，仍可采用冻结系数法，利用 $I+1/2$ 点的特征矩阵 $\mathbf{S}_{I+1/2}$ 将周围几个点（本程序中使用 $I-1, I, I+1$ 三个点）

上的守恒变量通过（2.5）式变换为特征变量 \mathbf{V}_I 。使用特征变量的好处是各物理量可尽量避免相互干扰，减少数值振荡。

本软件支持使用原始变量、守恒变量及特征变量来重构 $I+1/2$ 点上的物理量。对于每种方法，用户均可从 NND, WENO, UCD, MUSCL 四种格式中选择一种进行重构（见公式 (2.1)-(2.4)）。

例如，用户选择 2 阶 NND 格式重构 $I+1/2$ 点左值的公式为：

$$q_{I+1/2}^L = q_I + 1/2 \min \text{mod}(q_{I+1} - q_I, q_I - q_{I-1}) \quad (\text{原始变量重构})$$

$$U_{I+1/2}^L = U_I + 1/2 \min \text{mod}(U_{I+1} - U_I, U_I - U_{I-1}) \quad (\text{守恒变量重构})$$

$$V_{I+1/2}^L = V_I + 1/2 \min \text{mod}(V_{I+1} - V_I, V_I - V_{I-1}) \quad (\text{特征变量重构})$$

然后，可利用 \mathbf{U} 与 \mathbf{q} 的关系式或 \mathbf{U} 与 \mathbf{V} 的关系式（ $\mathbf{U} = \mathbf{S}^{-1} \mathbf{V}$ ）计算出 $I+1/2$ 点的 \mathbf{U} 。

§1.3 粘性项的离散

3.1 粘性通量的表达式

方程 (1.2) 最后一项为粘性项，其表达式如下：

$$\frac{1}{\Omega_{ij}} \oint_{\partial\Omega} \mathbf{F}_v \cdot \mathbf{n} ds = \frac{1}{\Omega_{ij}} (\mathbf{H}_{I+1/2,J}^v - \mathbf{H}_{I-1/2,J}^v + \mathbf{H}_{I,J+1/2}^v - \mathbf{H}_{I,J-1/2}^v) \quad (1.3.1)$$

穿过控制体右表面粘性通量的表达式为

$$H_{i+1/2,j}^v = [n_1 F_1^v + n_2 F_2^v]_{I+1/2,J} \Delta s_{I+1/2,J} \quad (1.3.2)$$

$$F_1^v = \begin{bmatrix} 0 \\ \tau_{11} \\ \tau_{12} \\ k \frac{\partial T}{\partial x} + u\tau_{11} + v\tau_{12} \end{bmatrix} \quad F_2^v = \begin{bmatrix} 0 \\ \tau_{12} \\ \tau_{22} \\ k \frac{\partial T}{\partial y} + u\tau_{12} + v\tau_{22} \end{bmatrix} \quad (1.3.3)$$

$$\tau_{ij} = \frac{\mu}{\text{Re}} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \quad (1.3.4)$$

$$k = \frac{\mu c_p}{\text{Pr Re}}$$

从表达式中可以看出，只需计算出(I+1/2,J)点处的导数 $\frac{\partial u_i}{\partial x_j}, \frac{\partial T}{\partial x_j}$ ，就可计算出穿过右表面的粘性通量。穿过其他三个表面的粘性通量同样计算。

3.2 导数的计算方法

在 (3.4) - (3.5) 中，需要计算物理量 ϕ 的导数 $\frac{\partial \phi}{\partial x}$ 和 $\frac{\partial \phi}{\partial y}$ (ϕ 为速度或温度)。计算方法有两种选择。

方法 1： 利用控制体上的 Gauss 公式

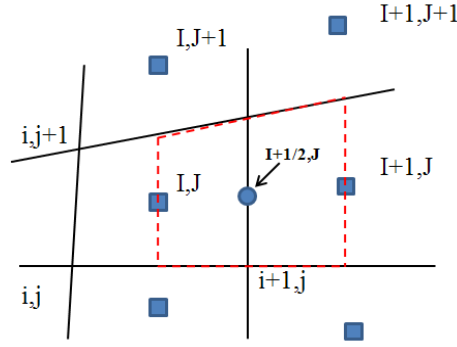


图 1.6 计算导数的控制体（红色虚线）

利用以 $(I+1/2, J)$ 为中心的控制体（图 1.6 中红色虚线围成）的高斯积分公式，可得到该导数的近似值。根据高斯公式：

$$\oint_{\partial\Omega_{I+1/2,J}} \varphi \mathbf{n} ds = \iint_{\Omega_{I+1/2,J}} \nabla \varphi d\Omega$$

假设控制体内梯度均匀分布，有：

$$\nabla \varphi \approx \frac{1}{\Omega_{I+1/2,J}} \oint_{\partial\Omega_{I+1/2,J}} \varphi \mathbf{n} ds$$

利用中点公式将积分展开，即可得 $(I+1/2, J)$ 点的导数值。

方法 2： 利用 Jocabian 变换

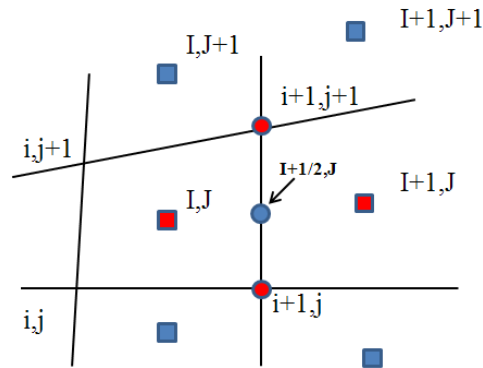


图 1.7 示意图：利用周围点信息计算 $(I+1/2, J)$ 点的导数值

在 **有限差分法** 中，通常利用 Jocabian 变换计算曲线坐标系下的导数值，有限体积法也可同样计算。计算公式为：

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial \xi} \xi_x + \frac{\partial \phi}{\partial \eta} \eta_x \quad (1.3.5)$$

$$\frac{\partial \phi}{\partial y} = \frac{\partial \phi}{\partial \xi} \xi_y + \frac{\partial \phi}{\partial \eta} \eta_y \quad (1.3.6)$$

其中 Jacobian 系数的计算方法为：

$$\begin{aligned} \xi_x &= Jy_\eta \\ \xi_y &= -Jx_\eta \\ \eta_x &= -Jy_\xi \\ \eta_y &= Jx_\xi \end{aligned}$$

$$J = \left| \frac{\partial(x, y)}{\partial(\xi, \eta)} \right|^{-1} = (x_\xi y_\eta - x_\eta y_\xi)^{-1} \quad (1.3.7)$$

利用 $(I+1/2, J)$ 周围 4 个点（图 3.2 中红色的点）的信息，可以计算出上式中的量：

$$\begin{aligned} \frac{\partial \psi}{\partial \xi} &= \psi_{I+1, J} - \psi_{I, J} \\ \frac{\partial \psi}{\partial \eta} &= \psi_{i+1, j+1} - \psi_{i+1, j} \end{aligned} \quad (1.3.8)$$

其中 ψ 可以是函数 ϕ （速度或温度），也可以是坐标 x 或 y 。

$(i+1, j)$ 及 $(i+1, j+1)$ 点上的坐标值是已知的，而这些点上的函数值可用周围格心点上的函数值插值获得。

利用（3.6）-（3.9）即可计算 $(I+1/2, J)$ 点上的导数值。**通过对比发现，该方法得到的公式与方法 1 完全相同。**

§1.4 网格的多块对接

为了处理更为复杂的几何形状，本软件支持多块对接网格。该网格是由多块网格对接而成，每块网格均为（相对光滑）的结构网格。

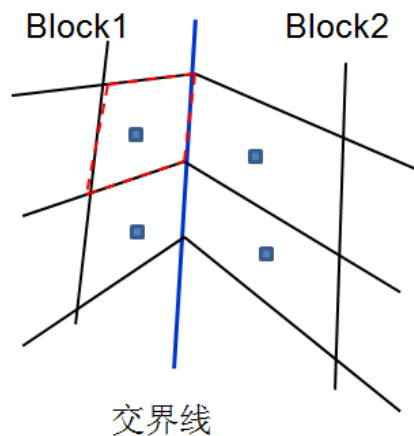


图 1.8 对接网格示意图

对接网格如图 1.8 所示，对接线上的计算与流体内点的计算方式相同。需要指出的是，由于对接线两侧的网格光滑性通常无法保证，因此对接线上通常采用 2 阶格式计算。如采用高阶格式，需考虑网格的非光滑性，形式比较复杂。

在程序中，采用“虚网格点”法，实现不同区域之间的数据交换。设流场的网格点为 $i=1,2,\dots,N_x$, $j=1,2,\dots,N_y$ ；定义物理量在 $i=0,\dots,N_x+1$, $j=0,\dots,N_y+1$ 。其中 $i=0$ 及 N_x+1 ； $j=0$ 及 N_y+1 的点为“虚网格点”，其物理采用相邻区域内的信息。

虚网格点上的信息根据网格连接文件（bc2d_1.in）的指示，从相邻 Block 中获取。

网格连接信息文件 bc2d_1.in 示例如下：

(算例： 30P30N 三段翼型绕流)

2D BC file for multi-block mesh: 30P30N

```
# Blocks
        60                                总块数
Block    1                                第 1 块
Subface
    4                                该块的子面数
f_no, face, istart,iend, jstart, jend,  neighb, subface orient
    1    1    1    1    1    65  -20    0    0
    2    4    1  129    65    65  -20    0    0
    3    2    1  129    1     1   17    3    2
    4    3  129  129    1    65    2    2    2
Block        2
Subface
    4
f_no, face, istart,iend, jstart, jend,  neighb, subface orient
    1    4    1   33    65    65  -20    0    0
    2    1    1    1    1    65    1    4    2
    3    2    1   33    1     1   18    3    2
    4    3   33   33    1    65    3    3    2
Block        3
Subface
    4
f_no, face, istart,iend, jstart, jend,  neighb, subface orient
    1    4    1   33    65    65  -20    0    0
    2    2    1   33    1     1   19    2    2
    3    1    1    1    1    65    2    4    2
    4    3   33   33    1    65    4    5    2
Block        4
Subface
    5
```

f_no	face	istart	iend	jstart	jend	neighb	subface	orient
1	4	17	41	65	65	-20	0	0
2	4	1	17	65	65	-20	0	0
3	2	1	41	1	1	20	3	2
4	3	41	41	1	65	5	2	2
5	1	1	1	1	65	3	4	2

.....

f_no: 子边界号

face: 子边界类型: 1 i-; 2 j-; 3 i+; 4 j+

istar, iend: 起始、终止的 i 值;

jstar, jend 起始、终止的 j 值;

neighb: 相邻的 Block 号

subface: 相邻的子边界号

ori: 连接方向 (2 顺序连接; 1, 3 逆序连接)

§1.5 特殊点的处理

1. 角点的处理

计算粘性通量时，“角点”需要特殊处理。图 1.9 为计算 $(1/2,1)$ 点处粘性通量时，使用的单元信息示意图。由于点对点连接信息交换的特点， $(0,0)$ 单元处的信息没有定义。因为块-块之间只交换相邻线两侧带状区域的信息，作为“角点”，该单元的值没有定义。图 1.10 为一个块的内点及虚网格点使用图，可以看出，在块-块对接的数据交换机制下，“角区”虚网格的值是没有定义的。如果采用多次信息交换，虽然可以获得这些角区的信息，但较为复杂（尤其是在并行系统中，多次信息传递对计算效率有一定影响）。

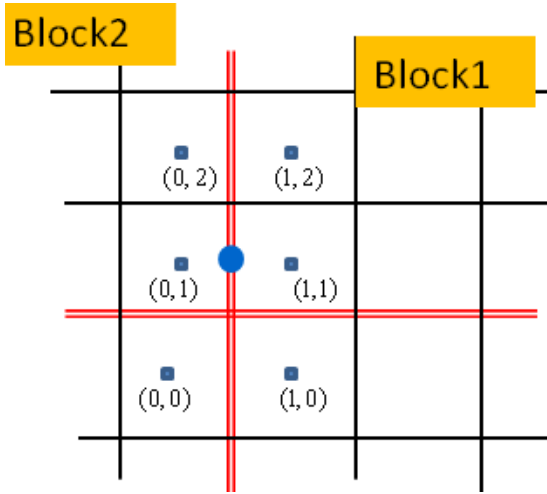


图 1.9 计算粘性通量使用的单元信息示意图

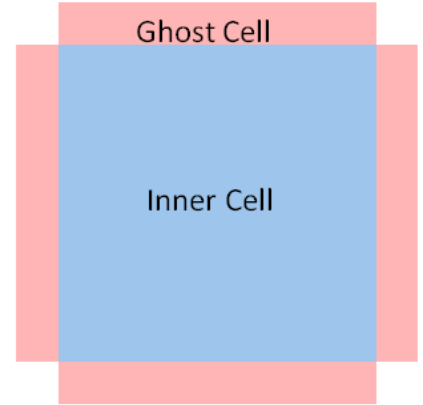


图 1.10 虚单元示意图

计算无粘通量时，无需使用角点信息，而计算粘性通量时，需要使用角点信息。如果角点信息没有定义，则需要特殊处理。

方法 1：用相邻内点的值近似

$$\left(\frac{\partial \phi}{\partial \eta} \right)_{1/2,1} \approx \frac{1}{2\Delta\eta} (\phi_{1/2,2} - \phi_{1/2,0}) \approx \frac{1}{2\Delta\eta} (\phi_{1,2} - \phi_{1,0})$$

方法 2：忽略该值

$$\left(\frac{\partial \phi}{\partial \eta} \right)_{1/2,1} \approx 0$$

实际上，这种方法是在角点处采用了“薄层近似”。

第二章 三维求解器

§2.1 基本方程

控制方程为三维守恒型 Navier-Stokes 方程

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_1(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{F}_2(\mathbf{U})}{\partial y} + \frac{\partial \mathbf{F}_3(\mathbf{U})}{\partial z} = \frac{\partial \mathbf{F}_{v1}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{F}_{v2}(\mathbf{U})}{\partial y} + \frac{\partial \mathbf{F}_{v3}(\mathbf{U})}{\partial z} \quad (2.1.1)$$

在以控制体上，对 N-S 方程进行积分得到如下方程

$$\frac{\partial \bar{\mathbf{U}}_{IJK}}{\partial t} + \frac{1}{\Omega_{IJK}} \oint_{\partial \Omega} \mathbf{F} \cdot \mathbf{n} ds - \frac{1}{\Omega_{IJK}} \oint_{\partial \Omega} \mathbf{F}_v \cdot \mathbf{n} ds = 0 \quad (2.1.2)$$

左端第 1 项为时间导数项，第 2 项为无粘项，第 3 项为粘性项。
本程序采用**网格中心型**控制体。

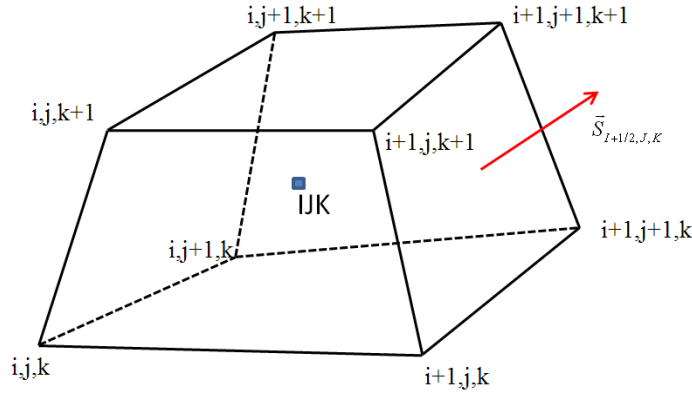


图 2.1 以 (IJK) 点为中心的控制体

1.1 控制体体积、表面积、表面法方向及中心点的计算公式

(Blazek et al. : Computational Fluid Dynamics, Principles and Applications, section 5.1.2)

侧面积及法向量计算公式为：

$$\vec{S}_{I+1/2,J,K} = \frac{1}{2} (\vec{r}_{i+1,j+1,k} - \vec{r}_{i+1,j,k+1}) \times (\vec{r}_{i+1,j+1,k+1} - \vec{r}_{i+1,j,k})$$

$$\vec{S} = S \vec{n}$$

控制体积的计算公式为：

$$\Omega = \frac{1}{3} \sum_m (\vec{r}_c \cdot \vec{S})_m$$

其中下标 m 表示该六面体的侧面。

控制体中心点的计算公式为：

$$\vec{r}_c = \frac{3 \sum_m (\vec{r}_c \cdot \vec{S})_m \vec{r}_{cm}}{4 \sum_m (\vec{r}_c \cdot \vec{S})_m}$$

§2.2 无粘项的离散

积分方程（2.1.2）中，无粘项为：

$$\frac{1}{\Omega_{IJK}} \oint_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} ds = \frac{1}{\Omega_{IJK}} \sum_m \mathbf{F}_m \cdot \vec{n}_m S_m$$

因此，关键是计算各侧面的流通量。

以(I+1/2,J,K)面的通量为例，说明无粘通量的计算步骤：

step 1: 重构，计算出 (I+1/2,J,K) 点处的流体变量的左、右值： $U_{I+1/2}^L, U_{I+1/2}^R$ 。

本软件使用的数值方法（NND,3 阶迎风，MUSCL,WENO3）均为三点格式。左值 $U_{I+1/2}^L$

利用 (I-1,J,K), (I,J,K), (I+1,J,K) 三个点上的值构造；右值 $U_{I+1/2}^R$ 利用 (I,J,K), (I+1,J,K),

(I+2,J,K) 三个点上的值构造。具体公式见 (1.2.6) - (1.2.9)。用户可选择使用原始变量、守恒变量或特征变量进行重构。详见第 1.2.5 节。

Step 2: 利用 FVS 或 FDS 方法，根据 $U_{I+1/2}^L, U_{I+1/2}^R$ 计算出穿过(I+1/2,J,K)点所在界面的通量。

FVS 方法的计算公式为： $F = F^+(U_{I+1/2}^L) + F^-(U_{I+1/2}^R)$ 。本软件使用 Steger-Warming FVS 方法，具体公式见 傅德薰《计算空气动力学》4.7.2 节。

FDS 是通过求解 Riemann 问题获知穿过界面的通量。由于使用精确 Riemann 解（Godnov 方法）计算量较大，本软件使用近似 Riemann 解。包括 HLL/HLLC 及 Roe 近似 Riemann 解。具体公式见 Toro : Riemann solvers and numerical methods for fluid dynamics, 第 10 章及第 11 章。也可参考作者的“计算流体力学”课件。

利用坐标旋转，将三维 Riemann 问题转化成界面法向的“扩展一维 Riemann 问题”（见作者的课件），以简化计算。以二维问题为例，示意图如下。

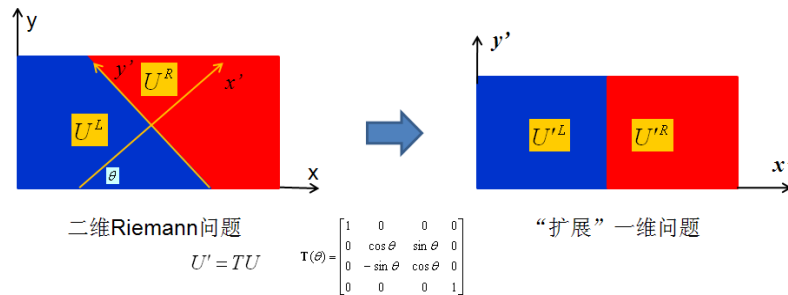


图 2.3 示意图：通过坐标旋转，将二维 Riemann 问题转化为“扩展”一维问题

具体步骤如下：

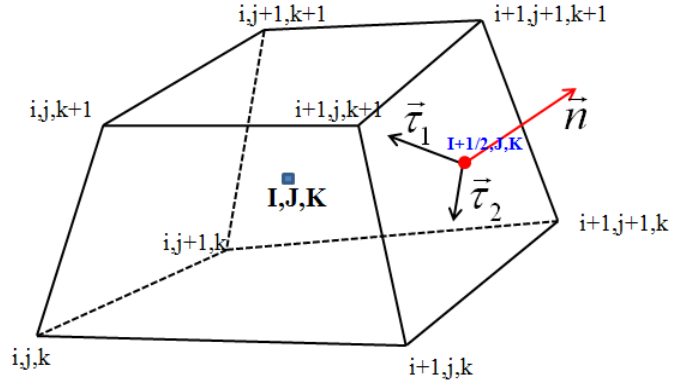


图 2.4 控制体及界面的法、切方向

a) 计算出 $(I+1/2, J, K)$ 所在界面的法方向和切方向（见图 2.1）:

$$\vec{n} = (\vec{r}_{i+1,j+1,k+1} - \vec{r}_{i+1,j,k}) \times (\vec{r}_{i+1,j,k+1} - \vec{r}_{i+1,j+1,k})$$

$$\vec{\tau}_1 = \vec{r}_{i+1,j+1,k+1} - \vec{r}_{i+1,j,k}$$

$$\vec{\tau}_2 = \vec{n} \times \tau_1$$

注: $\vec{n}, \vec{\tau}_1, \vec{\tau}_2$ 均为单位向量, 按上式计算后, 需单位化。

b) 通过坐标变换, 将通过 Step 1 得到的 $U_{I+1/2}^L, U_{I+1/2}^R$ 转化到 $(\vec{n}, \vec{\tau}_1, \vec{\tau}_2)$ 坐标系下。守恒变量 U 从原始坐标系到 $(\vec{n}, \vec{\tau}_1, \vec{\tau}_2)$ 坐标系的变换公式如下:

$$U' = \begin{pmatrix} \rho \\ \rho u' \\ \rho v' \\ \rho w' \\ E \end{pmatrix} \quad (2.1.3)$$

其中:

$$u' = u_n = \vec{u} \cdot \vec{n}, v' = u_{\tau_1} = \vec{u} \cdot \vec{\tau}_1, w' = u_{\tau_2} = \vec{u} \cdot \vec{\tau}_2 \quad (2.1.4)$$

分别为速度在新坐标系三个坐标轴方向的分量。（2.1.3）显示, 由于密度 ρ 和总能量（密度） E 是标量, 在坐标旋转过程中保持不变。动量（密度） $\rho \vec{u}$ 是矢量, 坐标变换过程中遵循旋转公式(2.1.4)。

将 $U_{I+1/2}^L, U_{I+1/2}^R$ 带入(2.1.3)-(2.1.4)式, 即可得到新坐标系中的守恒变量 $U_{I+1/2}^{\prime L}, U_{I+1/2}^{\prime R}$ 。

c) 在界面法方向 \vec{n} 求解扩展一维 Riemann 问题, 即可得到流通量 $(\vec{n}, \vec{\tau}_1, \vec{\tau}_2)$ 坐标系下的

值) $F'_{l+1/2}$ 。这里所说的“扩展”一维问题指, 问题本身是一维的(所以变量仅在 \vec{n} 方向变化, 在其他两个方向均匀分布), 但物理量是“扩展的”。因为纯一维问题的基本物理量是 $(\rho, \rho u', E)^T$, 而本问题的物理量是 $(\rho, \rho u', \rho v', \rho w', E)^T$, 多了两个切向动量分量。在求解 Riemann 问题中, “扩展”一维问题与原一维 Riemann 问题的解法相同, 切向分量的性质与**被动标量**相同, 不会给求解带来任何不便。具体公式见 Toro 书的第 10 及 11 章。

d) 得到流通量后, 为了便于和 (I,J,K) 点周围其他几个界面的流通量累加以求出进入该控制体积的总流通量, 需通过坐标旋转, 将 $F'_{l+1/2}$ 变回原先的 (x,y,z) 坐标系。令

$$\mathbf{F} = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \end{pmatrix}, \quad \mathbf{F}' = \begin{pmatrix} F'_1 \\ F'_2 \\ F'_3 \\ F'_4 \\ F'_5 \end{pmatrix}$$

分别表示 (x,y,z) 及 $(\vec{n}, \vec{\tau}_1, \vec{\tau}_2)$ 坐标系下的流通量。

则:

$$F_1 = F'_1, F_5 = F'_5$$

$$F_2 = F'_2 n_x + F'_3 \tau_{1x} + F'_4 \tau_{2x}$$

$$F_3 = F'_2 n_y + F'_3 \tau_{1y} + F'_4 \tau_{2y}$$

$$F_4 = F'_2 n_z + F'_3 \tau_{1z} + F'_4 \tau_{2z}$$

上式显示, 质量通量 F_1 及能量通量 F_5 是标量, 不随坐标旋转而变化。而动量通量 F_2, F_3, F_4 是矢量的分量, 随坐标旋转而变化。

Step 3: 将计算出的 (I+1/2,J,K) 点的流通量 $F_{l+1/2}$ 乘以该界面面积, 即得到穿过该界面的

流通量: $F_{l+1/2} S_{l+1/2}$ 。

采用同样方法, 可求出 (I,J+1/2,K) 及 (I,J,K+1/2) 所在界面的流通量。这样, 计算无粘流通量就完成了。

§2.3 粘性项的计算

1) 控制方程

$$\frac{1}{\Omega_{IJK}} \oint_{\partial\Omega} \mathbf{F}_v \cdot \mathbf{n} ds = \frac{1}{\Omega_{IJK}} \sum_m (\mathbf{F}_{vx} n_x + \mathbf{F}_{vy} n_y + \mathbf{F}_{vz} n_z)_m S_m \quad (2.3.1a)$$

其中：

$$\begin{aligned} F_{vx} &= \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k \frac{\partial T}{\partial x} \end{pmatrix} & F_{vy} &= \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k \frac{\partial T}{\partial y} \end{pmatrix} \\ F_{vz} &= \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k \frac{\partial T}{\partial z} \end{pmatrix} \end{aligned} \quad (2.3.1b)$$

其中 $k = \mu C_p / \text{Pr}$ 为（无量纲）导热系数。 C_p 为定压比热（无量纲），在本程序的无量

纲约定下，该系数为： $C_p = \frac{1}{(\gamma-1)Ma_\infty^2}$ 。 Pr 为 Prantdl 数，对于理想空气，通常取 $\text{Pr}=0.77$ 。

（无量纲）粘性系数的计算可采用如下 Sutherland 公式：

$$\mu = \frac{1}{\text{Re}} \left(\frac{\tilde{T}}{288.15} \right)^{3/2} \frac{288.15 + 110.4}{\tilde{T} + 110.4} \quad (2.3.2)$$

其中 $\tilde{T} = T\tilde{T}_\infty$ 为有量纲温度。

$$\boldsymbol{\tau} = \mu \begin{pmatrix} \frac{4}{3} \frac{\partial u}{\partial x} - \frac{2}{3} \left(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) & \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & \frac{4}{3} \frac{\partial v}{\partial y} - \frac{2}{3} \left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} \right) & \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} & \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} & \frac{4}{3} \frac{\partial w}{\partial z} - \frac{2}{3} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \end{pmatrix} \quad (2.3.3)$$

为粘性应力张量。

2) 偏导数的计算

从上段的公式可以看出,计算粘性项的关键在于偏导数 $\frac{\partial u_i}{\partial x_j}$ 及 $\frac{\partial T}{\partial x_j}$ 。计算出导数后,

就可求出粘性应力张量及粘性通量。

与二维程序相同,可使用 **Gauss** 公式,通过将体积分转化为面积分来计算导数。也可通过 **Jacobian** 变换来计算导数。二者得到的公式完全相同。写成 **Jacobian** 变换的形式更为简洁。**Jacobian** 变换是将物理空间 (x,y,z) 变换到均匀网格的计算空间 (ξ,η,ζ) 。

由于计算空间采用均匀网格,不妨假设计算空间网格间距为 1,这样 $(\xi,\eta,\zeta) = (I,J,K)$ 。及计算空间的坐标值就是网格的索引(下标)值。

在网格点 $(I+1/2,J,K)$ 上计算导数的具体公式如下:

$$\begin{aligned}\left.\frac{\partial \phi}{\partial x}\right|_{I+1/2,J,K} &= \left(\frac{\partial \phi}{\partial \xi} \xi_x + \frac{\partial \phi}{\partial \eta} \eta_x + \frac{\partial \phi}{\partial \zeta} \zeta_x\right)_{I+1/2,J,K} \\ \left.\frac{\partial \phi}{\partial y}\right|_{I+1/2,J,K} &= \left(\frac{\partial \phi}{\partial \xi} \xi_y + \frac{\partial \phi}{\partial \eta} \eta_y + \frac{\partial \phi}{\partial \zeta} \zeta_y\right)_{I+1/2,J,K} \\ \left.\frac{\partial \phi}{\partial z}\right|_{I+1/2,J,K} &= \left(\frac{\partial \phi}{\partial \xi} \xi_z + \frac{\partial \phi}{\partial \eta} \eta_z + \frac{\partial \phi}{\partial \zeta} \zeta_z\right)_{I+1/2,J,K}\end{aligned}\quad (2.3.4)$$

其中:

$$\begin{aligned}\left(\frac{\partial \phi}{\partial \xi}\right)_{I+1/2,J,K} &= \phi_{I+1,J,K} - \phi_{I,J,K} \\ \left(\frac{\partial \phi}{\partial \eta}\right)_{I+1/2,J,K} &= \phi_{I+1/2,J+1/2,K} - \phi_{I+1/2,J-1/2,K} \\ \left(\frac{\partial \phi}{\partial \zeta}\right)_{I+1/2,J,K} &= \phi_{I+1/2,J,K+1/2} - \phi_{I+1/2,J,K-1/2}\end{aligned}\quad (2.3.5)$$

这里 ϕ 为任意函数,可以是速度 u,v,w 、温度 T ,也可以是坐标 x,y,z 。

物理量(速度、温度)定义在格心,及整数 (I,J,K) 点。半点处的值可以用整点处的值插值得到(本程序使用)。例如:

$$\phi_{I+1/2,J+1/2,K} = \frac{1}{4}(\phi_{I,J,K} + \phi_{I+1,J,K} + \phi_{I,J+1,K} + \phi_{I+1,J+1,K}) \quad (2.3.6)$$

这种插值方法当网格接近平行四边形时具有二阶精度,当网格严重扭曲时为一阶精度。当网格扭曲严重时,如仍需二阶精度,需用更复杂的双线性插值(任玉新《计算流体力学基础》5.5.2 节)。

(2.3.4) 中变换系数的计算公式为（见傅德薰《计算空气动力学》 28 页）：

$$\begin{aligned}\xi_x &= J(y_\eta z_\zeta - z_\eta y_\zeta), \xi_y = J(z_\eta x_\zeta - x_\eta z_\zeta), \xi_z = J(x_\eta y_\zeta - y_\eta x_\zeta) \\ \eta_x &= J(y_\zeta z_\xi - z_\zeta y_\xi), \eta_y = J(z_\zeta x_\xi - x_\zeta z_\xi), \eta_z = J(x_\zeta y_\xi - y_\zeta x_\xi) \\ \zeta_x &= J(y_\xi z_\eta - z_\xi y_\eta), \zeta_y = J(z_\xi x_\eta - x_\xi z_\eta), \zeta_z = J(x_\xi y_\eta - y_\xi x_\eta)\end{aligned}\quad (2.3.7)$$

其中 Jocabian 行列式为：

$$J = \left| \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} \right|^{-1} = \begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{vmatrix}^{-1} \quad (2.3.8)$$

物理坐标 (x, y, z) 对计算坐标（即下标）的导数同样采用（2.3.5）式计算。例：

$$\begin{aligned}\left(\frac{\partial x}{\partial \xi} \right)_{I+1/2, J, K} &= x_{I+1, J, K} - x_{I, J, K} \\ \left(\frac{\partial x}{\partial \eta} \right)_{I+1/2, J, K} &= x_{I+1/2, J+1/2, K} - x_{I+1/2, J-1/2, K}\end{aligned}$$

实际计算中，需注意格心坐标与格点坐标的区别。本手册中，下标为大写字母的，表示格心值。下标为小写字母的，表示格点值（见图 2.4）。

【小结】 计算物理量偏导数的步骤：

- 1) 利用（2.3.5）式计算出各物理量（速度、温度、物理坐标）对计算坐标（下标）的偏导数： $u_\xi, u_\eta, u_\zeta, \dots, T_\xi, T_\eta, T_\zeta, x_\xi, x_\eta, x_\zeta, y_\xi, y_\eta, y_\zeta, \dots$
- 2) 利用(2.3.7)-(2.3.8)式，计算出 Jocabian 变换系数；
- 3) 利用（2.3.4）式计算出速度、温度对物理坐标 (x, y, z) 的偏导数。

计算出各偏导数后，带入（2.3.3）即可计算出粘性应力。带入（2.3.1）即可求出粘性通量。

§2.4 块的边界条件及连接信息

对于多块结构网格，需要对每个块的边界进行描述。以便明确其边界条件。指明其为物理边界还是与其他块相连的内边界。OpenCFD-EC 支持两种边界描述文件格式，Gridgen 的标准边界文件格式(.inp)以及 BXCFLD 的边界描述格式(.in)

2.4.1 Gridgen 标准边界描述文件格式 (xxx.inp)

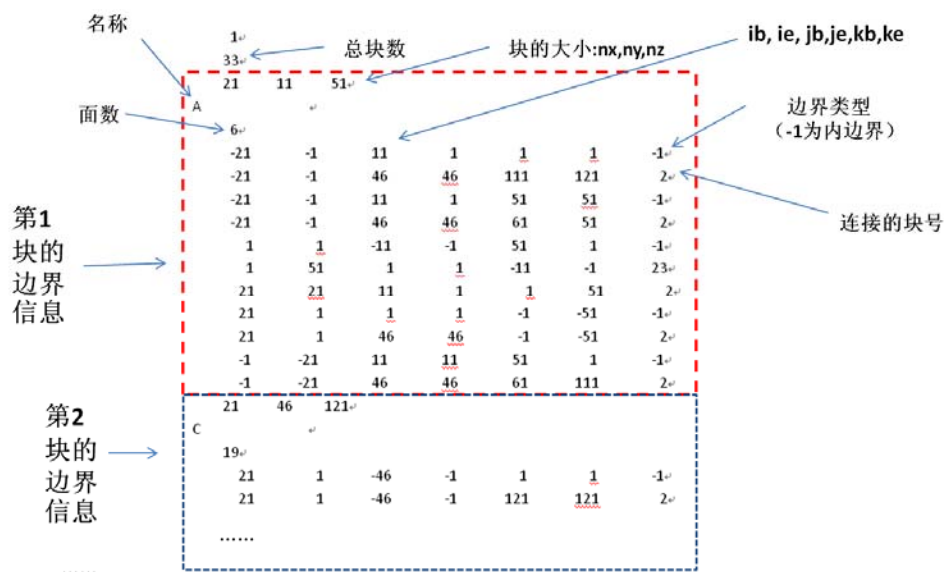


图 1 .inp 文件示例

上图为一个 Gridgen 标准边界描述文件（xxx.inp）的示例。其中第 2 行为总块数。其后为各块的边界条件描述。

每个块描述内容为：

- 1) 块的大小 nx,ny,nz;
- 2) 块的名称;
- 3) 边界面的数目;
- 4) 各个边界面的描述，形如：
ib, ie, jb, je, kb, ke, bc

表示该面的元素为 $ib \leq i \leq ie; jb \leq j \leq je; kb \leq k \leq ke$ 的区域。

例如上图第 1 个边界面为 下标 i 从 1 到 21； 下标 j 从 1 到 11； 下标 k=1 的区域（称为一个“子面”）。 bc 为该子面的边界类型。

bc=2 为固壁，bc=3 为对称面（或滑移壁面）；bc=4 为远场。

bc=-1 为内边界（即与其他块连接的虚边界，非物理边界）。

当该边界为内边界(bc=-1)时，该行的下一行为边界连接描述符。

网格连接说明：

图 1 所示的.inp 文件的第 1 对连接面为：

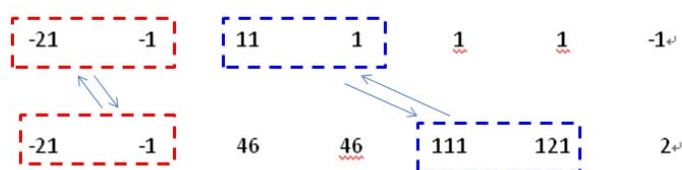


图 2 一对连接面

用符号（正或负）标示出连接的对象。如图 2 所示，正与正连接，负与负连接。图 2 所示的连接点为

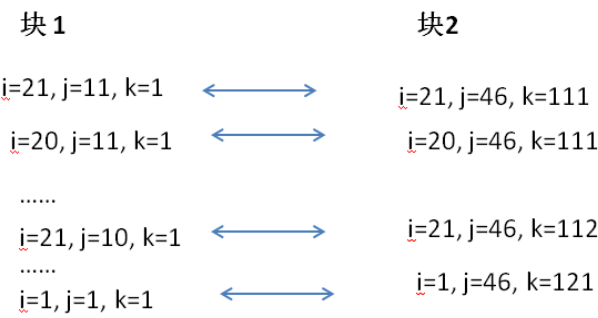
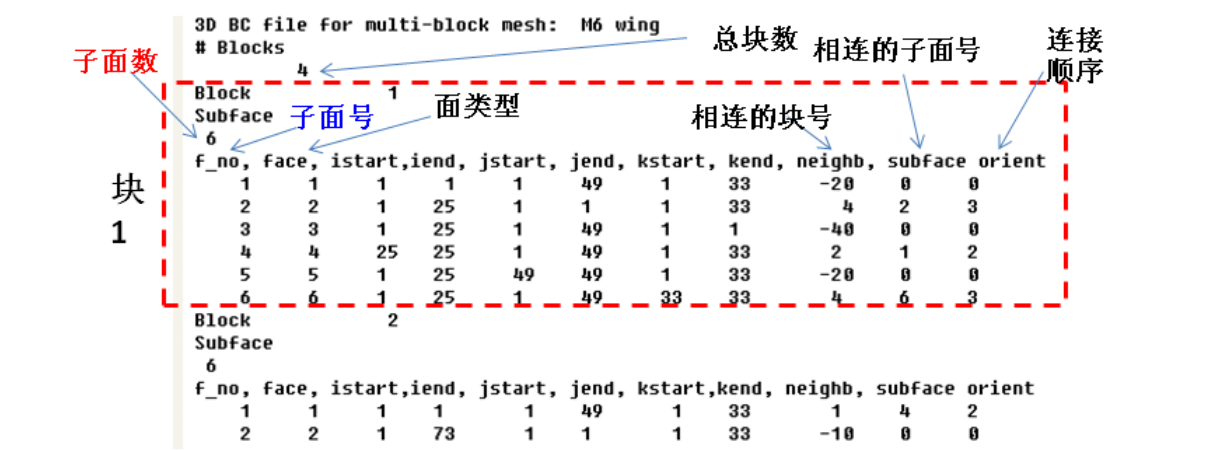


图 3 网格点之间的连接关系

2.4.2 BXCFD 边界描述格式 (.in)

BXCFD 的边界描述格式(.in)与 Gridgen 的.inp 文件类似。但.in 文件剔除了.inp 文件中的一些冗余信息，因而更为精简。



上图为一个.in 文件示例。

对于每一个子面，用一串数字来描述其定义及连接情况。

f_no 为该子面的编号。

face 为该子面的类型。 1-6 分别表示 i-,j-,k-,i+,j+,k+面。 即 i=1 的面,j=1 的面, k=1 的面, i=imax 的面, j=jmax 的面, k=kmax 的面。

istart,iend,jstart,jend,kstart,kend 描述这个面的大小。

Neighb 为与该面连接的块号。如果 neighb<0 则表示该面没有连接面，即该面为一个物理边界。 **neighb=-10,-20,-30,-40 分别表示壁面，远场，周期，对称边界条件**。neighb=-22 为强制出口边界条件(边界值由内点外插)。

Subface 表示与该面相连的面的编号（如该面没有连接面，则该值为 0）。

Orient 为连接顺序。为 0-4 的值。如该面没有连接面（物理边界）则为 0。 1-4 表示

4 中连接顺序。

采用对接网格，块与块之间的连接如图 1.8 所示。三维情况下，考虑到旋转，数据之间的对应关系略显复杂。

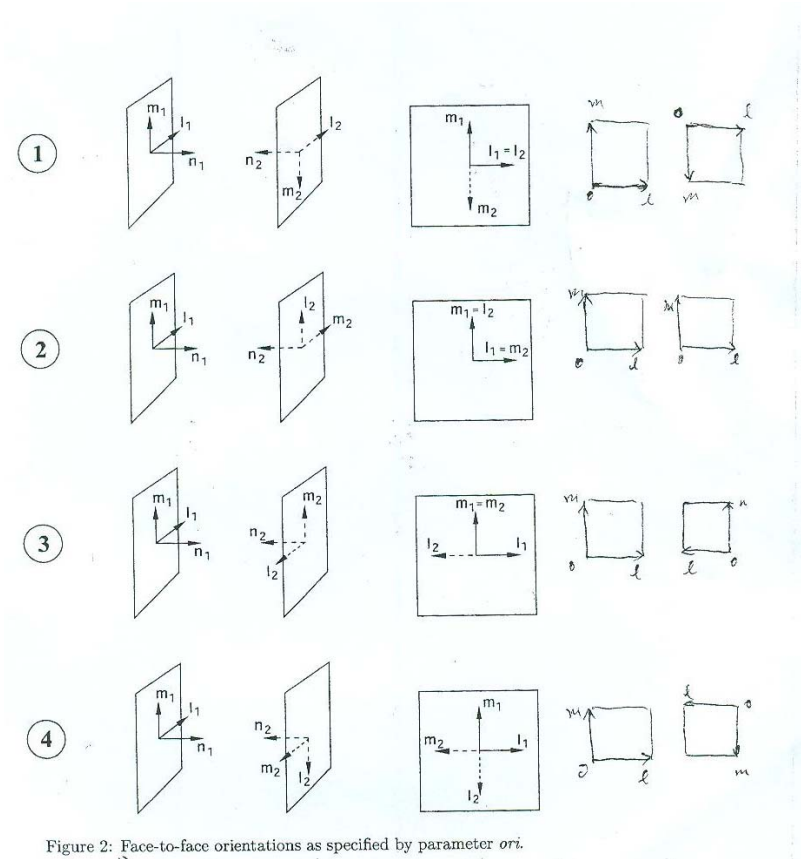


Figure 2: Face-to-face orientations as specified by parameter *ori*.

图 2.4.1 相邻块之间的对应关系

图 2.4.1 为块和块之间的对应关系示意图。其中 \vec{l}_1, \vec{m}_1 和 \vec{n}_1 为面 1 的两个切方向和法方向，他们之间的呈右手关系 $\vec{n}_1 = \vec{l}_1 \times \vec{m}_1$ ； \vec{l}_2, \vec{m}_2 和 \vec{n}_2 为面 2 的两个切方向和法方向，他们之间的呈右手关系 $\vec{n}_2 = \vec{l}_2 \times \vec{m}_2$ 。根据

由于块 B1 和块 B2 有共同的交界面，显然交界面处的外法线方向相反，即：

$$\vec{n}_1 = -\vec{n}_2$$

根据其中一个切方向的对应关系，即可确定整个交接面的对应关系。程序接口中，使用一个变量 *orient* 来表示这种对应关系，其含义如下：

表 1 $(\vec{l}_1, \vec{m}_1, \vec{n}_1)$ 与 $(\vec{l}_2, \vec{m}_2, \vec{n}_2)$ 的对应关系

Orient	含义
--------	----

1	$\vec{l}_2 = \vec{l}_1, \quad \vec{m}_2 = -\vec{m}_1, \quad \vec{n}_1 = -\vec{n}_2$
2	$\vec{l}_2 = \vec{m}_1, \quad \vec{m}_2 = \vec{l}_1, \quad \vec{n}_1 = -\vec{n}_2$
3	$\vec{l}_2 = -\vec{l}_1, \quad \vec{m}_2 = \vec{m}_1, \quad \vec{n}_1 = -\vec{n}_2$
4	$\vec{l}_2 = -\vec{m}_1, \quad \vec{m}_2 = -\vec{l}_1, \quad \vec{n}_1 = -\vec{n}_2$

1) 根据交界面的类型 (i-,j-,k-,i+,j+,k+) 确定沿下标增加的方向(i,j,k)与 (l,m,n)之间的对应关系。具体关系如下:

表 2 (l,m,n) 与 (i,j,k)的对应关系

类型	描述	(l,m)	备注
1 (i-)	n=-i, l=k, m=j	(2,1) (即 <i>l</i> 是该面的第 2 个下标, <i>m</i> 是第 1 个下标)	$\vec{k} \times \vec{j} = -\vec{i}$
2 (j-)	n=-j, l=i, m=k	(1,2)	$\vec{i} \times \vec{k} = -\vec{j}$
3 (k-)	n=-k, l=j, m=i	(2,1)	
4 (i+)	n=i, l=j, m=k	(1,2)	
5 (j+)	n=j, l=k, m=i	(2,1)	
6 (k+)	n=k, l=i, m=j	(1,2)	

处理方法如下:

- 1) 根据 ori 的值, 确定 $(\vec{l}_1, \vec{m}_1, \vec{n}_1)$ 与 $(\vec{l}_2, \vec{m}_2, \vec{n}_2)$ 的对应关系 (表 1)
- 2) 根据面的指向, 确定(i,j,k)与 (l,m,n) 的对应关系
- 3) 根据 1), 2) 确定 $(\vec{i}_1, \vec{j}_1, \vec{k}_1)$ 与 $(\vec{i}_2, \vec{j}_2, \vec{k}_2)$ 的对应关系。

【例】

面 1 的指向为 3 (k-), 面 2 的指向为 6 (k+); 二者的对应关系为 ori=2. 试确定两个面节点下标的对应关系。

查表 1 得:

$$l_1 = j_1, m_1 = i_1, \quad l_2 = i_2, m_2 = j_2 \quad (1)$$

查表 1 得：

$$l_2 = m_1, m_2 = l_1 \quad (2)$$

将 (1) 带入 (2) 得：

$$i_2 = i_1, j_2 = j_1$$

2.4.3 OpenCFD-EC 内建的边界信息格式

Opencfd-EC 3D ver 0.83 及以前的版本内建支持 BXCDF .in 格式，代码中利用 Block_TYPE 结构变量储存 BXCDF .in 格式的信息。对于 Gridgen 的 .inp 文件，由于子程序将其转化为 .in 文件。

从 0.84 版本开始，OpenCFD-EC 内建支持 Gridgen .inp 格式，同时也可读取 .in 格式(通过格式转化)。

OpenCFD-EC 读取 .inp 文件后，在存储过程中对其格式进行修改，以提高代码效率。

例如，bc3d.inp 中的一条连接信息为：

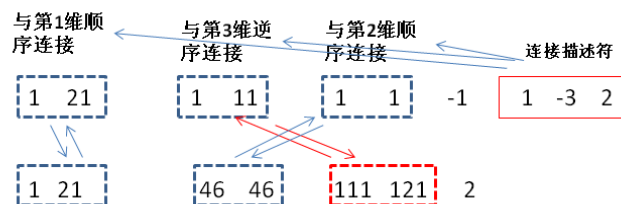
ib	ie	jb	je	kb	ke	
-21	-1	11	1	1	1	-1
-21	-1	46	46	111	121	2

则在 OpenCFD.in 中存储为：

1	21	1	11	1	1	-1	1	-3	2
1	21	46	46	111	121	2			

第 1 行最后三个数 (L1,L2,L3)=(1, -3, 2) 表示连接次序。

L1, L2 与 L3 分别表示该块第 1,2,3 维的连接情况。其取值 1,2,3 表示该维与对应块的第 1 维 (i), 第 2 维 (j) 或第 3 维 (k) 连接。正值表示顺序连接，负值表示逆序连接。示意图如下：



OpenCFD-EC 内部的连接格式虽然比 .inp 格式 **多了一些冗余信息(多了 L1,L2,L3 三个连接顺序描述符)**，但使用起来更为方便。ib,ie,jb,je,kb,ke 均为正数，且保证了 $ib \leq ie$ ， $jb \leq je$ ， $kb \leq ke$ 。从而 **避免了程序中繁琐的逻辑判断**。

对于无连接情况（物理边界），设定 $L1=L2=L3=0$ 。

此外，OpenCFD-EC 中 BC_MSG_TYPE 保留了 face 变量，描述该面的类型。1-6 分别表示

$i-, j-, k-, i+, j+, k+$ 面。这个变量为边界条件的处理及内边界连接的处理带来的方便。

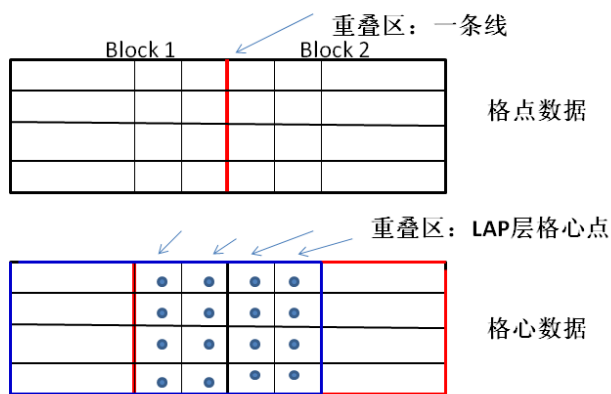


图 8 对接网格：格点数据的对接、格心数据的对接

为了保证内边界的计算精度，OpenCFD-EC 采用了 LAP 层虚边界进行对接（如上图所示）当前版本的有限体积模块最多支持 3 阶重构(MUSCL3),因而设置 LAP=2。如采用高阶重构方法，需要更改 LAP 值。例如，如采用 5 阶 WENO 格式，需设置 LAP=3，如采用 7 阶 WENO 格式，需要设置 LAP=4。

第三章 时间推进及加速收敛技术

§3.1 显格式（略）

§3.2 LU-SGS 隐格式

工程问题的 Reynolds 数通常很高，而湍流边界层的粘性尺度非常小。如果强制近壁处法向网格间距为 1 个粘性尺度 ($y^+ = 1$)，则网格尺度非常之小（通常只有计算域尺度的 10^{-7} - 10^{-8} 倍，甚至更小）。如果采用显格式进行时间推进，受稳定性制约，时间步长也需非常小，推进效率很低。即使采用局部时间步长法，计算效率仍显偏低。与显格式相比，隐式时间推进法可大幅放大时间步长，对提升计算效率有较大帮助。LU-SGS 具有稳定性好，格式简便，效率高等优点，是目前采用最多的隐式时间推进方法。从 ver 0.74 版本开始，本软件的三维代码支持 LU-SGS 方法。

3.2.1 求解线性代数方程组的 LU-SGS 方法

LU-SGS 方法本身是求解多对角带状系数矩阵线性方程组的一种迭代方法。以一个五对角的线性代数方程组为例，说明 LU-SGS 方法的原理。设方程组为：

$$a_{ij}^{(1)} x_{i-1,j} + a_{ij}^{(2)} x_{i,j-1} + a_{ij}^{(3)} x_{i,j} + a_{ij}^{(4)} x_{i+1,j} + a_{ij}^{(5)} x_{i,j+1} = b_{ij} \quad (3.2.1)$$

则可将该方程组近似分解为如下的两步

$$a_{ij}^{(1)} \bar{x}_{i-1,j} + a_{ij}^{(2)} \bar{x}_{i,j-1} + a_{ij}^{(3)} \bar{x}_{i,j} = b_{ij} \quad (3.2.2)$$

$$a_{ij}^{(3)} x_{i,j} + a_{ij}^{(4)} x_{i+1,j} + a_{ij}^{(5)} x_{i,j+1} = a_{ij}^{(3)} \bar{x}_{i,j} \quad (3.2.3)$$

(3.2.2) 和 (3.2.3) 分别为 **上三角** 方程组及 **下三角** 方程组。均可简单递推求解。实际上，将 (3.2.2) 与 (3.2.3) 相加，当迭代收敛时（即 $\bar{x}_{ij} = x_{ij}$ ）时，即可得到 (3.2.1)。从 (3.2.1) 分解为 (3.2.2)(3.2.3) 的方法称为近似 LU 分解法，写成矩阵形式即： $D + L + U \approx (D + L)D^{-1}(D + U)$ 。

通过简单递推，即可求解 (3.2.2) 及 (3.2.3)。例如 (3.2.2) 的递推公式为：

$$\bar{x}_{i,j} = (b_{ij} - a_{ij}^{(1)} \bar{x}_{i-1,j} - a_{ij}^{(2)} \bar{x}_{i,j-1}) / a_{ij}^{(3)} \quad (3.2.4)$$

显然，其递推过程为下标 (i,j) 从 (1,1) 向 (M,N) 的扫描过程（称为向上扫描，见图 3.2）。即从左下角向右上角扫描。同理 (3.2.4) 可通过从右上角向左下角扫描来递推求解。求解代数方程 (3.2.2) 需要多次扫描，直到收敛为止。

从以上过程来看，该迭代实际上是近似 LU 分解及对称 Gauss-Seidel 迭代的一种组合，因而被称为 LU 分解的对称 Gauss-Seidel 迭代(LU-SGS)。该方法提出时曾被称为 LU 分解的超松弛迭代法(LU-SOR),但方法本身并不含松弛因子，因而称为 LU-SGS 更为恰当。

但在 Euler（或 N-S）方程的 LU-SGS 方法中，每次迭代 **仅需一对扫描过程**（向上、向下扫描各一次）即可，无需在达到收敛。因为时间推进收敛时，扫描过程自然会达到收敛。

$$\bar{x}_{11} \rightarrow \bar{x}_{12}, \bar{x}_{21} \rightarrow \bar{x}_{31}, \bar{x}_{22}, \bar{x}_{13} \rightarrow \dots$$

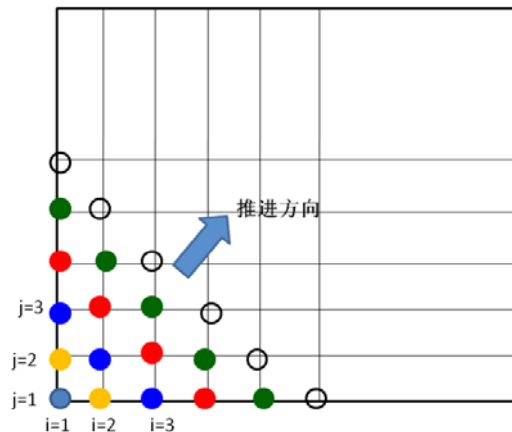


图 3.2 LU-SGS 扫描过程示意图

3.2.2 求解 Euler 方程的 LU-SGS 方法

1) 方程的离散

以 Euler 方程为例，阐述 LU-SGS 方法的基本思想，该方法很容易推广到 N-S 方程。

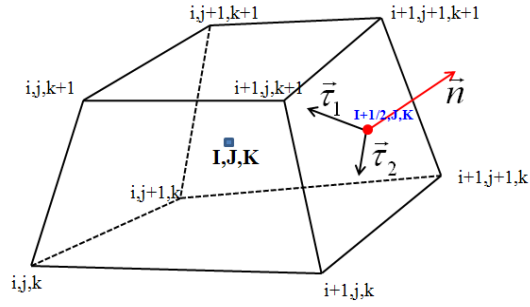


图 3.3 控制体示意图

在图 3.3 所示的控制体上，考虑三维积分形式的 Euler 方程：

$$\Omega_{IJK} \frac{\partial \bar{U}_{IJK}}{\partial t} + \oint_{\partial\Omega} \mathbf{F}_n ds = 0 \quad (3.2.5)$$

其中 $\mathbf{F}_n = \mathbf{F} \cdot \mathbf{n} = \mathbf{F}_1 n_1 + \mathbf{F}_2 n_2 + \mathbf{F}_3 n_3$ 为控制体界面法向的流通量。

采用一阶 Euler 隐格式对时间进行离散，得

$$\Omega_{IJK} \frac{\bar{U}^{n+1} - \bar{U}^n}{\Delta t} + \oint_{\partial\Omega} \mathbf{F}_n^{n+1} ds = 0 \quad (3.2.6)$$

在 (3.2.6) 两端同时加上

$$-\oint_{\partial\Omega} \mathbf{F}_n^n ds \triangleq \mathbf{R}_{IJK}^n \quad (3.2.7)$$

显然 \mathbf{R}_{IJK}^n 就是 n 时刻的残差（右端项），可以用前文介绍的方法显式计算出来。定义：

$\mathbf{W}^n \triangleq \Delta \bar{U}^n \triangleq \bar{U}^{n+1} - \bar{U}^n$ ，则 (3.2.6) 化为

$$\Omega_{IJK} \frac{\mathbf{W}^n}{\Delta t} + \oint_{\partial\Omega} (\mathbf{F}_n^{n+1} - \mathbf{F}_n^n) ds = \mathbf{R}_{IJK}^n \quad (3.2.8)$$

其中 $\mathbf{W}^n = \Delta \mathbf{U}$ 是需要计算的物理量，该物理量隐式给出，需要进行方程组求解。求解出该物理量后，通过 $\bar{U}^{n+1} = \bar{U}^n + \mathbf{W}^n$ 就可以计算出 n+1 步物理量的值。需要指出的是，

对于**定常问题**，当时间推进收敛时， $\mathbf{W}^n \rightarrow 0$ ，因而（3.2.8）式**左端的离散并不影响物理量 \mathbf{U} 的精度**。物理量 \mathbf{U} 的空间精度由(3.2.8)式右端项 \mathbf{R}^n 的计算方法决定，而 \mathbf{R}^n 可由(3.2.7)显式计算，其计算方法见本手册第二章。对于非定常问题，可采用双时间步长法，在伪时间步采用 LU-SGS 方法。

由于：

$$\mathbf{F}_n^{n+1} - \mathbf{F}_n^n \approx \left(\frac{\partial \mathbf{F}_n}{\partial \mathbf{U}} \right)^n (\mathbf{U}^{n+1} - \mathbf{U}^n) = \mathbf{A}_n^n \mathbf{W}^n$$

则（3.2.8）可化为：

$$\Omega_{IJK} \frac{\mathbf{W}^{n+1}}{\Delta t} + \oint_{\partial\Omega} \mathbf{A}^n \mathbf{W}^{n+1} ds = \mathbf{R}_{IJK}^n \quad (3.2.9)$$

其中 $\mathbf{A} = \frac{\partial \mathbf{F}_n}{\partial \mathbf{U}} = \frac{\partial}{\partial \mathbf{U}} (\mathbf{F}_1 n_1 + \mathbf{F}_2 n_2 + \mathbf{F}_3 n_3)$ 为法向通量的 Jacobian 矩阵。

采用**一阶迎风格式**计算（3.2.9）中各界面处的通量。

其中，穿过 $I+1/2$ 界面的通量为：

$$(\mathbf{A}\mathbf{W})_{I+1/2} = [(\mathbf{A}^+ + \mathbf{A}^-)\mathbf{W}]_{I+1/2} = \mathbf{A}_I^+ \mathbf{W}_I + \mathbf{A}_{I+1}^- \mathbf{W}_{I+1} \quad (3.2.10)$$

为了简单，可以采用 L-F 分裂方法

$$\mathbf{A} = \mathbf{A}^+ + \mathbf{A}^-; \mathbf{A}^\pm = (\mathbf{A} \pm \omega \lambda_A^{\max} \mathbf{I}) / 2 \quad (3.2.11)$$

其中 $\lambda_A^{\max} = \max(|u_n|, |u_n + c|, |u_n - c|) = |u_n| + c$ 为矩阵 \mathbf{A} 的最大特征值(即该矩阵的“谱半径”)。

$\omega \in [1, 2]$ **为松弛因子**，增加该值会提高迭代的稳定性，但会降低收敛速度。 $u_n = \vec{V} \cdot \vec{n}$ 为控制体界面上的法向速度。

于是：

$$\begin{aligned} & \oint_{\partial\Omega} \mathbf{A}_n^n \mathbf{W}^{n+1} ds \\ &= (\mathbf{A}_I^+ \mathbf{W}_I + \mathbf{A}_{I+1}^- \mathbf{W}_{I+1}) S_{I+1/2} - (\mathbf{A}_{I-1}^+ \mathbf{W}_{I-1} + \mathbf{A}_I^- \mathbf{W}_I) S_{I-1/2} \\ &+ (\mathbf{A}_J^+ \mathbf{W}_J + \mathbf{A}_{J+1}^- \mathbf{W}_{J+1}) S_{J+1/2} - (\mathbf{A}_{J-1}^+ \mathbf{W}_{J-1} + \mathbf{A}_J^- \mathbf{W}_J) S_{J-1/2} \\ &+ (\mathbf{A}_K^+ \mathbf{W}_K + \mathbf{A}_{K+1}^- \mathbf{W}_{K+1}) S_{K+1/2} - (\mathbf{A}_{K-1}^+ \mathbf{W}_{K-1} + \mathbf{A}_K^- \mathbf{W}_K) S_{K-1/2} \end{aligned} \quad (3.2.12)$$

（3.2.9）离散为：

$$\begin{aligned}
& W_{JK} \left(\frac{\Omega}{\Delta t} \mathbf{I} + A_I^+ S_{I+1/2} - A_I^- S_{I-1/2} + A_J^+ S_{J+1/2} - A_J^- S_{J-1/2} + A_K^+ S_{K+1/2} - A_K^- S_{K-1/2} \right) \\
& + A_{I+1}^- W_{I+1} S_{I+1/2} - A_{I-1}^+ W_{I-1} S_{I-1/2} + A_{J+1}^- W_{J+1} S_{J+1/2} - A_{J-1}^+ W_{J-1} S_{J-1/2} + A_{K+1}^- W_{K+1} S_{K+1/2} - A_{K-1}^+ W_{K-1} S_{K-1/2} \\
& = R_{JK}^n
\end{aligned} \tag{3.2.13}$$

这就是物理量 \mathbf{W} 的控制方程，该方程为 9 对角方程线性代数组，可采用前文介绍的 LU-SGS 方法求解。

2) 离散方程的化简

虽然可直接采用 LU-SGS 等方法求解方程 (3.2.13)，但该方程仍稍显复杂，特别是包含了矩阵及矩阵求逆运算，对计算量及内存消耗较大。考虑到迭代本身就是一个近似过程，无需精确计算，因而可对其进行化简，以减少计算量。

简化 1: 对角线元素单位阵化

对于结构网格，由于 $S_{I+1/2} \approx S_{I-1/2}$ ，且两个界面的法方向也相差不大，因而

$$A_I^+ S_{I+1/2} - A_I^- S_{I-1/2} \approx S_I (A_I^+ - A_I^-) = S_I \omega \hat{\Lambda}_c^I \mathbf{I}$$

其中：

$$\hat{\Lambda}_c^I = \lambda_t^{\max} S_I = (|u_n^I| + c) S_I \tag{3.2.14}$$

$$u_n^I = \vec{V} \cdot \vec{n}^I, \quad \vec{n}^I = \frac{1}{2}(\vec{n}_{I+1/2} + \vec{n}_{I-1/2}), \quad S_I = (S_{I+1/2} + S_{I-1/2}) / 2$$

对于其余界面(J+1/2,J-1/2,K+1/2,K-1/2)同样处理，则 (3.2.13) 简化为：

$$\begin{aligned}
& W_{JK} \left(\frac{\Omega}{\Delta t} + \omega(\hat{\Lambda}_c^I + \hat{\Lambda}_c^J + \hat{\Lambda}_c^K) \right) \mathbf{I} + \\
& A_{I+1}^- W_{I+1} S_{I+1/2} - A_{I-1}^+ W_{I-1} S_{I-1/2} + A_{J+1}^- W_{J+1} S_{J+1/2} - A_{J-1}^+ W_{J-1} S_{J-1/2} + A_{K+1}^- W_{K+1} S_{K+1/2} - A_{K-1}^+ W_{K-1} S_{K-1/2} \\
& = R_{JK}^n
\end{aligned} \tag{3.2.14}$$

其中， $\hat{\Lambda}_c^J = (|u_n^J| + c) S_J$, $\hat{\Lambda}_c^K = (|u_n^K| + c) S_K$ ，其定义方法与 (3.2.14) 相似。

因九对角（块）方程组 (3.2.14) 的对角线上的元素为单位矩阵，因而采用 LU-SGS 时可避免繁琐的矩阵求逆计算。

简化 2: 简化矩阵计算

(3.2.14) 式虽然可避免矩阵求逆计算，但仍需计算诸如 $\mathbf{A}_{I+1}^- \mathbf{W}_{I+1}$ 之类的矩阵计算。虽然

Jacobian 矩阵 \mathbf{A}_{I+1}^- 并不复杂，但矩阵计算量仍比标量计算大很多。

由于: $\mathbf{A}_{I+1} \mathbf{W}_{I+1} = \frac{\partial \mathbf{F}_n}{\partial \mathbf{U}} \Delta \mathbf{U} \approx \Delta \mathbf{F}_n = \mathbf{F}_n(\mathbf{U}^n + \mathbf{W}) - \mathbf{F}_n(\mathbf{U}^n)$

因而, 矩阵计算可近似简化为如下标量计算:

$$\mathbf{A}_{I+1}^- \mathbf{W}_{I+1} S_{I+1/2} = \frac{1}{2} (\mathbf{A} S_{I+1/2} - \omega \hat{\Lambda}_c^I \mathbf{I})_{I+1} \mathbf{W}_{I+1} \approx \frac{1}{2} (\Delta \mathbf{F} - \omega \hat{\Lambda}_c^I \mathbf{W})_{I+1} \quad (3.2.15)$$

$$\Delta \mathbf{F} = \mathbf{F}_n(\mathbf{U}^{new}) - \mathbf{F}_n(\mathbf{U}^{old}) = \begin{pmatrix} \rho u_n \\ \rho u u_n + p n_x \\ \rho v u_n + p n_y \\ \rho w u_n + p n_z \\ (E + p) u_n \end{pmatrix}^{new} - \begin{pmatrix} \rho u_n \\ \rho u u_n + p n_x \\ \rho v u_n + p n_y \\ \rho w u_n + p n_z \\ (E + p) u_n \end{pmatrix}^{old}$$

由于 LU-SGS 的上扫过程中, 计算(I,J,K)点的 \mathbf{W} 时, (I-1,J,K), (I,J-1,K), (I,J,K-1)点的 \mathbf{W} 已被计算出来, 因而 $\mathbf{U}^{new} = \mathbf{U}^{old} + \mathbf{W}$ 已知, 故很容易计算 $\Delta \mathbf{F}$ 。下扫过程也是同样。

3) 求解过程

最终, LU-SGS 方法的计算过程如下

Step 1: 采用前文给出的方法显式计算出右端项 (残差) R_{IJK}^n

Step 2 进行向上扫描:

$$\bar{W}_{IJK} = (R_{IJK}^n + A_{I-1}^+ W_{I-1} S_{I-1/2} + A_{J-1}^+ W_{J-1} S_{J-1/2} + A_{K-1}^+ W_{K-1} S_{K-1/2}) / \alpha$$

$$\text{其中: } \alpha = \frac{\Omega}{\Delta t} + \omega(\hat{\Lambda}_c^I + \hat{\Lambda}_c^J + \hat{\Lambda}_c^K)$$

Step 3: 进行向下扫描:

$$W_{IJK} = (\alpha \bar{W}_{IJK} - A_{I+1}^- \bar{W}_{I+1} S_{I+1/2} - A_{J+1}^- \bar{W}_{J+1} S_{J+1/2} - A_{K+1}^- \bar{W}_{K+1} S_{K+1/2}) / \alpha$$

计算过程中, 可利用(3.2.15)式来简化矩阵计算。

Step 4: 时间推进

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \mathbf{W}$$

3.2.4 LU-SGS 方法向 N-S 方程的推广

上一节针对 Euler 方程介绍了 LU-SGS 方法。对于粘性方程, 原则上也可同样处理。积分型 N-S 方程为:

$$\Omega_{IJK} \frac{\partial \bar{\mathbf{U}}_{IJK}}{\partial t} + \oint_{\partial \Omega} \mathbf{F}_n ds = \oint_{\partial \Omega} \mathbf{F}_n^v ds \quad (3.2.16)$$

两端同时加上 n 时间步的残差

$$\oint_{\partial \Omega} (\mathbf{F}_n - \mathbf{F}_n^v)^n ds \triangleq \mathbf{R}_{IJK}^n$$

得

$$\Omega_{IJK} \frac{\mathbf{W}^n}{\Delta t} + \oint_{\partial\Omega} [(\mathbf{F}_n - \mathbf{F}_n^v)^{n+1} - (\mathbf{F}_n - \mathbf{F}_n^v)^n] ds = \mathbf{R}_{IJK}^n \quad (3.2.17)$$

其中 $\mathbf{W}^n \triangleq \Delta \bar{\mathbf{U}}^n \triangleq \bar{\mathbf{U}}^{n+1} - \bar{\mathbf{U}}^n$ 为两个时间步物理量的差量。

由于：

$$\mathbf{F}_n^{n+1} - \mathbf{F}_n^n \approx \mathbf{A}_n \mathbf{W}^n; \mathbf{F}_n^{v,n+1} - \mathbf{F}_n^{v,n} \approx \mathbf{A}_n^v \mathbf{W}^n$$

则 (3.2.17) 式化为：

$$\Omega_{IJK} \frac{\mathbf{W}^n}{\Delta t} + \oint_{\partial\Omega} [\mathbf{A}_n - \mathbf{A}_n^v] \mathbf{W}^n ds = \mathbf{R}_{IJK}^n \quad (3.2.18)$$

其中

$$\mathbf{A}_n = \frac{\partial \mathbf{F}_n}{\partial \mathbf{U}}; \mathbf{A}_n^v = \frac{\partial \mathbf{F}_n^v}{\partial \mathbf{U}}$$

分别为无粘和粘性通量的 Jacobian 矩阵。

在图 3.3 所示的控制体上，将 (3.2.18) 式离散化。对无粘通量采用一阶迎风差分离散（见上节）；**对粘性通量采用二阶中心差分离散。**

由于粘性项的 Jacobian 矩阵计算较为复杂，因而很多情况下可进行近似计算，即将该矩阵对角化。例如

$$\mathbf{A}^v \mathbf{S} \approx \hat{\Lambda}_v \mathbf{I}$$

其中：

$$\hat{\Lambda}_c = \max\left(\frac{4}{3\rho}, \frac{\gamma}{\rho}\right) \left(\frac{\mu}{\text{Pr}} + \frac{\mu_T}{\text{Pr}_T}\right) \frac{S^2}{\Omega}$$

该对角化过程类似于将 N-S 方程的粘性进行如下简化：

$$\frac{\partial \mathbf{F}_v}{\partial x} \approx \Lambda_v \frac{\partial^2 \mathbf{U}}{\partial x^2}$$

于是，方程 (3.18) 化为：

$$\begin{aligned} & W_{IJK} \left[\frac{\Omega}{\Delta t} + \omega(\hat{\Lambda}_c^I + \hat{\Lambda}_c^J + \hat{\Lambda}_c^K) + 2(\hat{\Lambda}_v^I + \hat{\Lambda}_v^J + \hat{\Lambda}_v^K) \right] \mathbf{I} + \\ & A_{I+1}^- W_{I+1} S_{I+1/2} - A_{I-1}^+ W_{I-1} S_{I-1/2} + A_{J+1}^- W_{J+1} S_{J+1/2} - A_{J-1}^+ W_{J-1} S_{J-1/2} + A_{K+1}^- W_{K+1} S_{K+1/2} - A_{K-1}^+ W_{K-1} S_{K-1/2} \\ & - \hat{\Lambda}_{v,I+1} W_{I+1} - \hat{\Lambda}_{v,I-1} W_{I-1} - \hat{\Lambda}_{v,J+1} W_{J+1} - \hat{\Lambda}_{v,J-1} W_{J-1} - \hat{\Lambda}_{v,K+1} W_{K+1} - \hat{\Lambda}_{v,K-1} W_{K-1} \\ & = R_{IJK}^n \end{aligned} \quad (3.2.19)$$

对 (3.2.19) 可采用采用 LU-SGS 方法，进行上-下扫描求解，方法与前文相同。

最终，N-S 方程的 LU-SGS 方法的计算过程如下

Step 1: 采用前文给出的方法显式计算出右端项（残差） R_{IJK}^n

Step 2 进行向上扫描：

$$\bar{W}_{IJK} = (R_{IJK}^n + A_{I-1}^+ W_{I-1} S_{I-1/2} + A_{J-1}^+ W_{J-1} S_{J-1/2} + A_{K-1}^+ W_{K-1} S_{K-1/2} + \hat{\Lambda}_{v,I-1} W_{I-1} + \hat{\Lambda}_{v,J-1} W_{J-1} + \hat{\Lambda}_{v,K-1} W_{K-1}) / \alpha \text{ 其中:}$$

$$\alpha = \frac{\Omega}{\Delta t} + \omega(\hat{\Lambda}_c^I + \hat{\Lambda}_c^J + \hat{\Lambda}_c^K) + 2(\hat{\Lambda}_v^I + \hat{\Lambda}_v^J + \hat{\Lambda}_v^K) \quad (3.2.20)$$

Step 3: 进行向下扫描：

$$W_{IJK} = (\alpha \bar{W}_{IJK} - A_{I+1}^- \bar{W}_{I+1} S_{I+1/2} - A_{J+1}^- \bar{W}_{J+1} S_{J+1/2} - A_{K+1}^- \bar{W}_{K+1} S_{K+1/2} + \hat{\Lambda}_{v,I+1} W_{I+1} + \hat{\Lambda}_{v,J+1} W_{J+1} + \hat{\Lambda}_{v,K+1} W_{K+1}) / \alpha$$

计算过程中，可利用(3.2.15)式来简化矩阵计算。

Step 4: 时间推进

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \mathbf{W}$$

§3.3 多重网格法

详细内容见 J. Blazek “Computational Fluid Dynamics, Principles and Applications”. Section 9.4.

1. 算法步骤

设空间离散后，控制方程为

$$\frac{d}{dt} \mathbf{W}_h = -\frac{1}{\Omega_h} \mathbf{R}_h \quad (3.1.1)$$

其中下标“h”表示最密的网格，下标“2h”表示网格粗化一次后的网格。

计算步骤如下：

1) 假设已知 n 时间步物理量的值 \mathbf{W}_h^n ，代入(3.1.1)式可计算出 n+1 时刻物理量的值 \mathbf{W}_h^{n+1} ，根据该值计算出残差 \mathbf{R}_h^{n+1} 。

2) 将物理量和残差插值到粗网格

$$\mathbf{W}_{2h}^{(0)} = I_h^{2h} \mathbf{W}_h^{n+1}$$

其中 I_h^{2h} 为从细网格到粗网格的插值算子。

3) 利用粗网格上的物理量 $\mathbf{W}_{2h}^{(0)}$ ，计算出粗网格上的残差 $\mathbf{R}_{2h}^{(0)}$

计算该残差与细网格上的残差（粗化后）之差：

$$(\mathbf{Q}_F)_{2h} = I_h^{2h} \mathbf{R}_h^{n+1} - \mathbf{R}_{2h}^{(0)}$$

该值的含义是一个修正量，即对粗网格残差的修正。该项称为“强迫函数”（forcing function），可防止粗网格计算带来的误差。

4) 在粗网格上进行时间推进求解

$$\frac{d}{dt} \mathbf{W}_{2h} = -\frac{1}{\Omega_{2h}} (\mathbf{R}_{2h} + (\mathbf{Q}_F)_{2h})$$

可推进一到数个子步。其中 $(\mathbf{Q}_F)_{2h}$ 在推进过程中保持不变。推进数个子步后，得到粗网格上的物理量 \mathbf{W}_{2h}^{n+1}

5) 计算粗网格上物理量的修正量

$$\delta W_{2h} = W_{2h}^{n+1} - W_{2h}^{(0)}$$

将该修正量插值到细网格，计算出细网格上物理量的新值

$$W_h^+ = W_h^{n+1} + I_{2h}^h \delta W_{2h}$$

需要指出的是，当计算收敛时（达到定常状态） δW_{2h} 趋近于 0，因此**粗网格上的计算精度并不影响细网格上的计算结果**。因此，粗网格的空间离散可用低精度方法计算（例如 1 阶迎风），以提高计算效率、增强计算稳定性以及快速耗散掉物理量的震荡。

2. 插值的具体方法

1) 细网格向粗网格上的插值

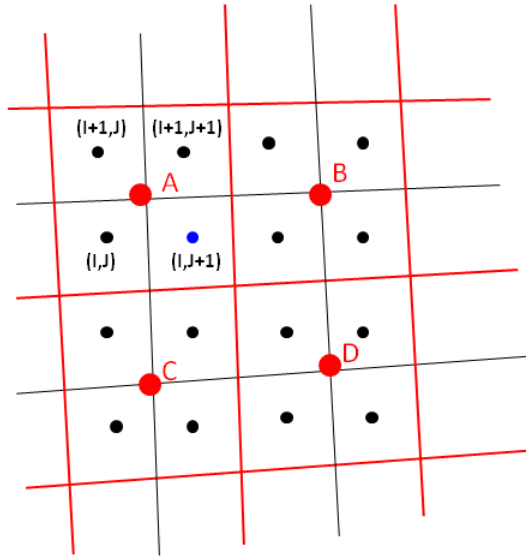


图 2.5.1 细网格（红线及黑线）与粗网格（红线）

如图 2.5.1 所示，红线为粗网格，红线+黑线为细网格。红点为粗网格上物理量所在位置，黑点为细网格上物理量所在的位置（均定义在网格中心）。

根据有限体积法中物理量及通量的定义，容易给出从细网格到粗网格上的插值公式：

$$W_A = \frac{\Omega_{i,j} W_{i,j} + \Omega_{i+1,j} W_{i+1,j} + \Omega_{i,j+1} W_{i,j+1} + \Omega_{i+1,j+1} W_{i+1,j+1}}{\Omega_{i,j} + \Omega_{i+1,j} + \Omega_{i,j+1} + \Omega_{i+1,j+1}} \quad (3.1.2)$$

这个公式是**精确成立**的，没有引入误差。因为有限体积法中 $W_{i,j}$ 的含义是区域 $\Omega_{i,j}$ 内的平均量，(3.1.2) 式显然是该物理量在大区域内的平均量。

残差的插值更为简单：

$$R_A = R_{i,j} + R_{i+1,j} + R_{i,j+1} + R_{i+1,j+1}$$

这个公式也是**精确成立**的。因为残差的物理含义是控制体内的净通量，将各区域净

通量相加，显然就是大区域的净通量。

对于三维情况，同样处理。即，粗网格上物理量的值为周围细网格物理量的加权平均（以细网格控制体体积为权重）。粗网格上的残差为周围细网格上残差的简单相加。

2) 粗网格向细网格上的插值

以点 $(I,J+1)$ 为例（图中蓝色的点）。该点周围有4个粗网格上的点A,B,C,D; $(I,J+1)$ 点上的值应当是这4个点值的加权平均。根据距离远近分配权重。在网格相对均匀的情况下，插值公式为：

$$W_{I,J+1} = \frac{9}{16}W_A + \frac{3}{16}W_B + \frac{3}{16}W_C + \frac{1}{16}W_D$$

即，距离该点最近的A点贡献了9/16的权重，次近的点B和C分别贡献了3/16的权重，最远的D点贡献了1/16权重。

对于三维情况细网格上的物理量是其周围8个粗网格物理量的加权平均。最近点的权重是27/64，次近的三个点的权重是9/64，再次近的三个点权重是3/64，最远点的权重是1/64。

在网格相对均匀的情况下，这种插值方式有**二阶精度**。

§3.4 双时间步长法

上文介绍的加速收敛技术，包括局部时间步长法、隐格式及多重网格法均针对定常问题。而对于非定常问题，可以借助双时间步长法，将上述加速收敛技术推广使用。

对于方程

$$\Omega \frac{\partial U}{\partial t} = R(U) \quad (3.4.1)$$

构造如下2阶精度隐格式：

$$\Omega \frac{3U^{n+1} - 4U^n + U^{n-1}}{2\Delta t} = R(U^{n+1}) \quad (3.4.2)$$

为了求解(3.4.2),构造如下推进方程：

$$\Omega \frac{\partial U^*}{\partial t^*} = R(U^*) - \Omega \frac{3U^* - 4U^n + U^{n-1}}{2\Delta t} \triangleq R^*(U^*) \quad (3.4.3)$$

从(3.4.3)可以看出，如果该式收敛（达到定常状态），则 $\frac{\partial U^*}{\partial t^*} = 0$ ，则 $U^* = U^{n+1}$ 。因而(3.4.3)可作为从 U^n 到 U^{n+1} 的内迭代， t^* 为内迭代时间（又称“伪时间”）。

双时间步长法的计算步骤为：

- 1) 给定初值 U^0 ；

- 2) 推进 1 个时间步, 得到 U^1 (例如, 可适应 Runge-Kutta 显格式);
 - 3) 假设已知 U^n 及 U^{n-1} , 求解方程 (3.4.3), 得到定常解, 即为 U^{n+1} ;
- 由于在步骤 3), 对于时间 t^* 为定常解, 因而可采用各种加速收敛技术, 例如局部时间步长、隐格式、多重网格等。

通常情况下, 伪时间步的推进采用隐格式 (例如 LU-SGS 格式), 具体如下:

$$\Omega \frac{\partial U^*}{\partial t^*} = R^*(U^*)$$

离散为:

$$\begin{aligned} \Omega \frac{U^{*(k+1)} - U^{*(k)}}{\Delta t^*} &= R^*(U^{*(k+1)}) \\ \Omega \frac{\delta U^*}{\Delta t^*} &= (R^*(U^{*(k+1)}) - R^*(U^{*(k)})) + R^*(U^{*(k)}) \\ &= \frac{\partial R^*}{\partial U^*} \delta U^* + R^*(U^{*(k)}) \\ &= -\frac{3}{2\Delta t} \Omega \delta U^* + \frac{\partial R}{\partial U} \delta U^* + R(U^{*(k)}) - \Omega \frac{3U^{*(k)} - 4U^n + U^{n-1}}{2\Delta t} \end{aligned}$$

与上文的(单时间步)LU-SGS 相比, 双时间 LU-SGS 格式变化并不大。
在程序实现上, 如果已有了单时间步 LU-SGS 格式, 只需进行如下简单修改即可:

- 1) 将对角线值由 α (见 3.2.20) 改为 $\alpha + \frac{3\Omega}{2\Delta t}$;
- 2) 将右端项改变为 $R(U^{*(k)}) - \Omega \frac{3U^{*(k)} - 4U^n + U^{n-1}}{2\Delta t}$

需要注意 LU-SGS 迭代中使用的时间步长为伪时间步长 Δt^* , 伪时间步长可采用局部时间步长; 而物理时间推进过程中采用的时间步长 Δt 不能为局部时间步长。

第四章 湍流模型

当 Reynolds 数足够高时，流动呈湍流状态。湍流的多尺度特性给计算带来了很大的挑战。湍流的计算方法包括直接数值模拟(DNS)，大涡模拟(LES)及湍流模式理论(RANS，又称 Reynolds 平均 NS 方程)等。其中 DNS 直接求解 NS 方程，不使用湍流模型，因而不含模型误差，准确度最高，但由于其必须分辨湍流的全部细节，因而计算量非常巨大，对于高 Reynolds 数的复杂工程问题，通常计算机无法承受。LES 对 NS 方程进行滤波，对大尺度量进行求解，而对小尺度量的影响采用模型（即亚格子模型，SGS 模型）。LES 的计算量虽然比 DNS 降低了很多，但对于复杂的工程问题，其计算量仍然难以承受。湍流模型理论(RANS)对湍流进行平均，只对平均量进行计算，脉动量的影响通过数学模型体现在控制方程中。由于只计算平均量，因而 RANS 的计算量较小，是目前计算复杂工程流动问题的主要方法。

RANS 需要对湍流提模型，目前尚没有能够精确计算（平均量）且普适的湍流模型。湍流模型研究是目前计算流体力学领域的研究热点和难点。

目前最常使用的湍流模型为涡粘模型，该模型假设 Reynolds 应力与平均应变率张量呈正比，其比例系数为湍流粘性系数。采用涡粘模型后，RANS 方程形式上与原 N-S 方程相同，仅仅是将原先的层流粘性系数 μ_L 替换成层流粘性系数+湍流粘性系数 $\mu_L + \mu_T$ ，将原先的层流热传导系数 $k_L = \mu_L Cp / Pr$ 替换成层流热传导系数+湍流热传导系数 $k_L + k_R = Cp(\mu_L / Pr_L + \mu_T / Pr_T)$ 即可。因而，湍流的影响归结为湍流粘性系数 μ_T 的计算。除了湍流粘性系数需要单独计算外，其余部分与原先的层流（NS 方程）的计算相同，前几章介绍的方法均可使用。OpenCFD-EC 支持的三种湍流模型：BL,SA 和 SST k-w 模型均为涡粘模型。

4.1 BL 模型

参考文献： D.C. Wilcox, *Turbulence Modeling for CFD*, DCW Industries

4.1.1 控制方程

BL 模型属于代数模型，无需求解偏微方程就可直接通过代数表达式给出湍流粘性系数。由于无需求解偏微方程，因而属于零方程模型。

BL 模型是混合长模型的改进模型，主要用于边界层层流动。其将流体分为“内层”和“外层”，在内层采用混合长模型，在外层采用尾迹亏损率模型。为了保证在壁面上湍流粘性系数为 0，在近壁区采用了指数型衰减函数。具体公式如下：

$$\mu_t = \begin{cases} \mu_{ti} & \text{if } y \leq y_c \\ \mu_{to} & \text{otherwise} \end{cases}$$

其中 y_c 是 $\mu_{ti} = \mu_{to}$ 是到壁面的最小距离。

内层模型：

$$\mu_{ii} = \rho l^2 \|\Omega\|$$

$$l = ky(1 - e^{-y^+/A^+})$$

其中 $\Omega = \nabla \times V$ 为涡量, $\|\Omega\| = \sqrt{\Omega_x^2 + \Omega_y^2 + \Omega_z^2}$ 为涡量的模。 $y^+ = \rho_w y U_\tau / \mu_w$, $U_\tau = \sqrt{\frac{\mu_w}{\rho_w} \frac{\partial u}{\partial y} \Big|_w}$,

y 为到壁面的距离。。

显然, 这是一个混合长模型, 在近壁区使用了衰减函数令湍流粘性系数衰减为 0.

外层模型:

$$\mu_{to} = \rho \alpha C_{cp} F_{wake} F_{Kleb}$$

$$F_{wake} = \min(y_{\max} F_{\max}, C_{wk} y_{\max} U_{dif}^2 / F_{\max});$$

$$F_{\max} = \frac{1}{k} \max_y (l |\Omega|),$$

$$F_{Kleb}(y) = \left[1 + 5.5 \left(\frac{C_{Kleb} y}{y_{\max}} \right)^6 \right]^{-1}$$

$$U_{dif} = U_{\max} - U_{y=y_{\max}}$$

$$U = \sqrt{u^2 + v^2 + w^2}$$

y_{\max} 为函数 $l|\Omega|$ 达到最大值时的 y 值。 注意到在对数率区 $l|\Omega|$ 基本保持常数值 (因为对

数率区 $u \sim \ln y$, $\|\Omega\| \approx \frac{\partial u}{\partial y} \sim 1/y$), 而在外层该值会衰减, 因而 y_{\max} 可以认为是对数率区外

缘的位置。 F_{Kleb} 为 Klebnoff 间歇函数, 该函数反应了边界层外层湍流的间歇性 (0 为层流, 1 为湍流), 该函数随着到壁面距离的增加, 从 0 衰减到 1。

模型系数为: $\alpha = 0.0168$, $C_{cp} = 1.6$, $C_{wk} = 1$, $C_{Kleb} = 0.3$, $A^+ = 26$ 。 $k = 0.4$ 为 Karman 常数。

4.1.2 计算方法

1) BL 模型需要沿壁面法方向进行搜索 (找到 F_{\max} 及 y_{\max})。但实际上, 为了计算方便, 经常用网格线代替垂直于法方向的直线。因而使用 BL 模型时, 近壁网格须具有较好的正交性。

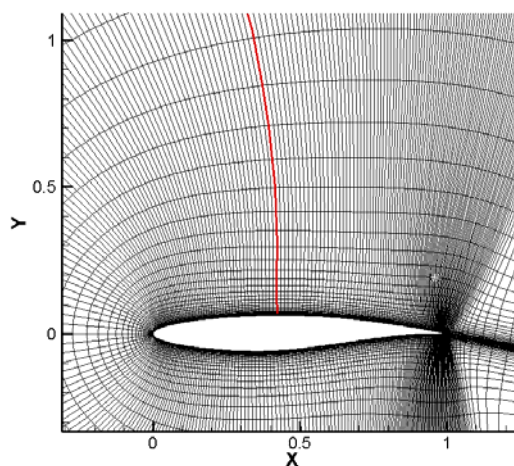


图 4.1 搜索 F_{\max}, y_{\max} 使用的网格线示意图

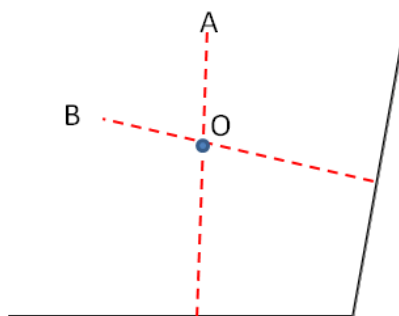


图 4.2 角点周围 BL 模型的使用

2) BL 模型适用于含有单个壁面的边界层。当某点周围有多个壁面时（例如角点周围），该模型较为复杂。本软件采用简化方法，湍流粘性系数取为沿两条线计算的湍流粘性系数的最小值。如图 4.2 所示，计算 O 点的湍流粘性系数，可沿网格线 A 计算，也可沿网格线 B 计算，这样就会产生两个湍流粘性系数。计算中，取二者的最小值。

3) BL 模型仅能给出壁面边界层周围的湍流粘性系数。对于没有壁面的情况（例如尾流区）需采用其他方法计算湍流粘性系数。本软件使用 BL 模型时，对于不含壁面的区域，湍流粘性系数默认为 0。这会带来一定误差，如对尾流区等不含壁面区域的流动感兴趣，建议采用其他模型。

4) BL 模型基于混合长模型，原则上只能应用与无分离或弱分离的流动。对于大范围分离流，该模型适用性较差。实际上，目前的湍流模型(RANS)对于大范围分离流的预测能力都不够理想，这些复杂流动可使用 DES, LES 甚至 DNS 来计算。

4.2 SA 模型

参考文献： J. Blazek, Computational Fluid Dynamics: Principles and Applications, Elsevier, section 7.2.1

<http://turbmodels.larc.nasa.gov/spalart.html> (本代码采用其中的形式)

4.2.1 控制方程

SA 模型是 Spalart 和 Allmaras 提出来的一种一方程模型。该模型直接假设湍流粘性系数满足流场中的标量方程（带源项的对流-扩散方程），并基于经验给出了方程及系数。该模型中包含大量的经验常数及经验函数。由于 Spalart 等人在航空计算方面具有丰富的经验并掌握丰富的实验（计算）数据，这些经验常数及经验函数经过了很好的调校，SA 模型在航空方面计算效果较好。

SA 模型给出的湍流粘性系数为：

$$\mu_t = f_{v1} \rho \tilde{\nu}$$

$$\frac{\partial \tilde{\nu}}{\partial t} + \frac{\partial}{\partial x_j} (\tilde{\nu} u_j) = C_{b1} (1 - f_{t2}) \tilde{S} \tilde{\nu} - \left[C_{w1} f_w - \frac{C_{b1}}{k^2} f_{t2} \right] \left(\frac{\tilde{\nu}}{d} \right)^2 + \frac{1}{\sigma} \left[\frac{\partial}{\partial x_j} \left((v + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right) + C_{b2} \frac{\partial \tilde{\nu}}{\partial x_j} \frac{\partial \tilde{\nu}}{\partial x_j} \right]$$

$\tilde{S} = \max(\|\Omega\| + \frac{\tilde{\nu}}{k^2 d^2} f_{v2}, 0.3 \|\Omega\|)$ （注：对 \tilde{S} 进行限制，防止其出现负值，造成计算不稳定）。 $\|\Omega\|$ 为涡量的模。

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad f_{t2} = C_{t3} \exp(-C_{t4} \chi^2), \quad \chi = \frac{\tilde{\nu}}{v}$$

$$f_w = g \left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right]^{1/6}, \quad g = r + C_{w2} (r^6 - r), \quad r = \min \left[\frac{\tilde{\nu}}{\tilde{S} k^2 d^2}, 10 \right]$$

$$C_{b1} = 0.1355, \sigma = 2/3, C_{b2} = 0.622, k = 0.41, C_{w2} = 0.3$$

$$C_{w3} = 2, C_{v1} = 7.1, C_{t3} = 1.2, C_{t4} = 0.5, C_{w1} = C_{b1} / k^2 + (1 + C_{b2}) / \sigma$$

边界条件为：

$$\text{壁面条件：} \quad \tilde{\nu}_{wall} = 0$$

$$\text{来流条件：} \quad \tilde{\nu}_{\infty} = 3v_{\infty} \sim 5v_{\infty}$$

出口条件：简单外推（令出口外虚网格点上的值等于内点的值）。

初始条件：设定 $\tilde{\nu} = \tilde{\nu}_{\infty}$

4.3.2 计算方法

在控制体内对 $\tilde{\nu}$ 的控制方程进行积分。对流项采用一阶迎风格式（采用 L-F 分裂）进行离散。界面处的对流通量为：

$$f_{j+1/2} = f_j^+ + f_{j+1}^- = \left[\frac{1}{2} (u_n + |u_n|)_j \tilde{\nu}_j + \frac{1}{2} (u_n - |u_n|)_{j+1} \tilde{\nu}_{j+1} \right]$$

粘性项采用二阶中心格式进行离散。采用下面等式，对其进行简化：

$$\frac{1}{\sigma} \left[\frac{\partial}{\partial x_j} \left((v + \tilde{v}) \frac{\partial \tilde{v}}{\partial x_j} \right) + C_{b2} \frac{\partial \tilde{v}}{\partial x_j} \frac{\partial \tilde{v}}{\partial x_j} \right] = \frac{1 + C_{b2}}{\sigma} \left[\frac{\partial}{\partial x_j} \left((v + \tilde{v}) \frac{\partial \tilde{v}}{\partial x_j} \right) \right] - \frac{C_{b2}(v + \tilde{v})}{\sigma} \frac{\partial^2 \tilde{v}}{\partial x_j \partial x_j}$$

上式右侧第一项的离散方法与 **N-S** 方程中粘性项的离散方式相同。右侧第二项采用如下方法在控制体上离散：

$$\iiint_{\Omega_{ijk}} \frac{C_{b2}(v + \tilde{v})}{\sigma} \frac{\partial^2 \tilde{v}}{\partial x_j \partial x_j} d\Omega \approx \frac{C_{b2}(v + \tilde{v})}{\sigma} \bigg|_{ijk} \iiint_{\Omega_{ijk}} \frac{\partial^2 \tilde{v}}{\partial x_j \partial x_j} d\Omega = \frac{C_{b2}(v + \tilde{v})}{\sigma} \bigg|_{ijk} \oint_S \frac{\partial \tilde{v}}{\partial x_j} n_j dS$$

源项：

$$Q = C_{b1}(1 - f_{t2})\tilde{S}\tilde{v} - \left[C_{w1}f_w - \frac{C_{b1}}{k^2}f_{t2} \right] \left(\frac{\tilde{v}}{d} \right)^2$$

可直接计算。其中涡量采用二阶中心差分进行计算。

SA 模型源项的刚性并不是很强，因而本程序并未对源项进行隐式处理。在采用 **LU-SGS** 隐格式计算时，**SA** 方程的源项仍显式计算。但由于 \tilde{v} 方程粘性项的系数是 **NS** 动量方程粘性项系数的 $1/\sigma = 3/2$ 倍，因而在 **LU-SGS** 中， \tilde{v} 方程的主对角线元素需要比动量方程的元素大些。其主对角线元素为：

$$\alpha^v = \frac{\Omega}{\Delta t} + \omega(\hat{\Lambda}_c^I + \hat{\Lambda}_c^J + \hat{\Lambda}_c^K) + 3(\hat{\Lambda}_v^I + \hat{\Lambda}_v^J + \hat{\Lambda}_v^K)$$

而动量方程 **LU-SGS** 离散主对角线元素为：

$$\alpha = \frac{\Omega}{\Delta t} + \omega(\hat{\Lambda}_c^I + \hat{\Lambda}_c^J + \hat{\Lambda}_c^K) + 2(\hat{\Lambda}_v^I + \hat{\Lambda}_v^J + \hat{\Lambda}_v^K)$$

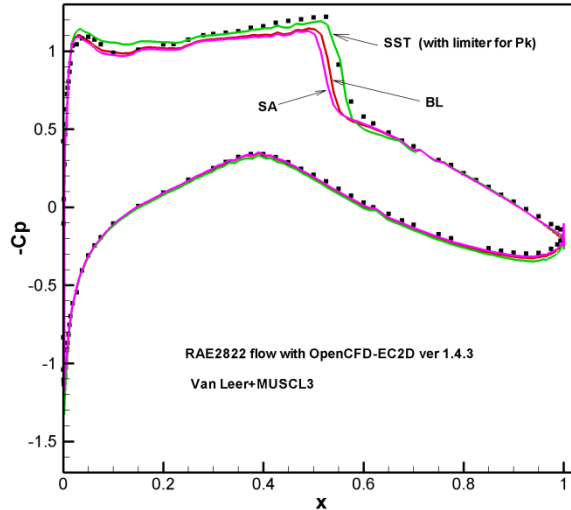


图 4.2.1 采用不同湍流模型计算的 RAE2822 翼型表面的压力系数分布

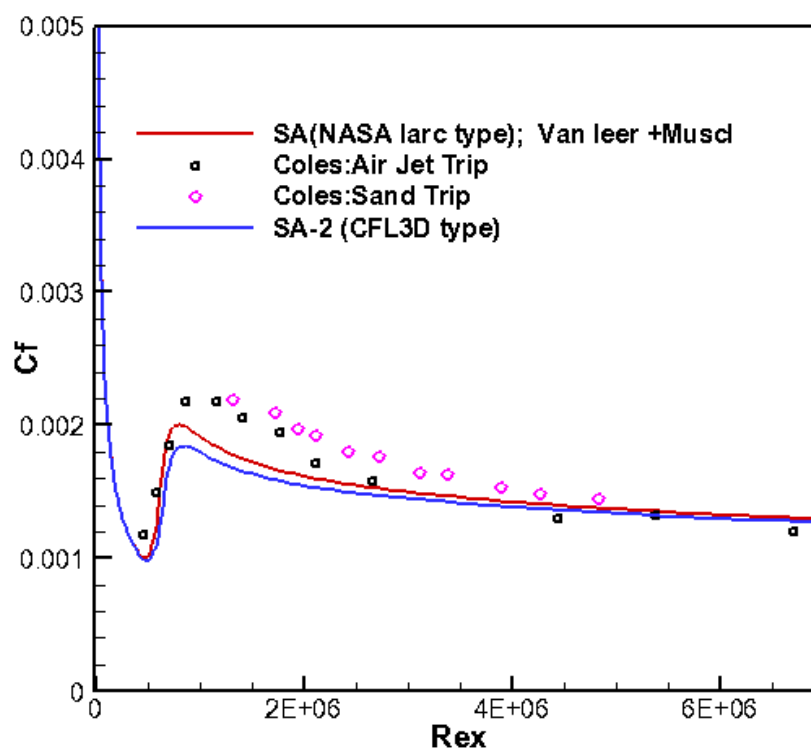
图 4.2.1 为采用不同湍流模型计算的 RAE2822 翼型表面的压力系数分布，从中可以看出，如果以背风面的激波为判据，**SA** 模型的计算结果不如 **BL** 及 **SST** 模型。

湍流模型的适用性相当复杂，一个算例并不能说明问题。建议用户针对不同问题对湍流模型进行测试（后验评估），从而总结出不同湍流模型的适应范围。

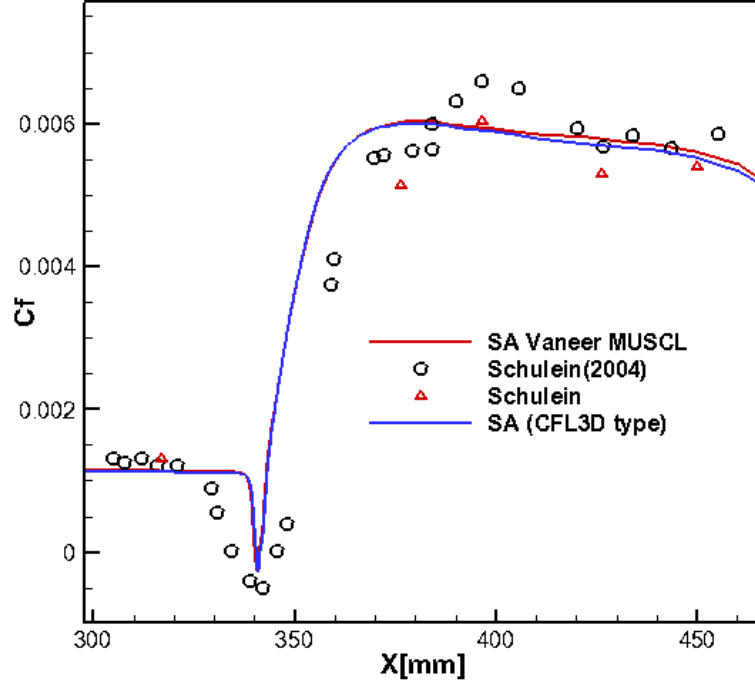
4.3.3 SA 模型的不同版本

自 Splart 等人提出 SA 模型以来，很多学者提出了修改版。这些修改版主要对 SA 源项进行了修改。本软件自 0.98c 版，采用了 CFL3D 中的版本。其源项为：

$$Q_{SA}^{CFL3D} = C_{b1}(1 - f_{t2})\tilde{S}\tilde{v} + \left[-C_{w1}f_w + \frac{C_{b1}}{k^2}[(1 - f_{t2})f_{v2} + f_{t2}] \right] \left(\frac{\tilde{v}}{d} \right)^2$$



平板边界层算例，摩擦阻力系数分布 ()



入射激波-湍流边界层几次干扰，摩擦阻力系数分布 ()

4.3 SST k- ω 模型

4.3.1 控制方程

该模型采用下式计算湍流粘性系数（见《CFL3D Version 5.0 manual》，Blazek et al. 7.2.3 节）：

（<http://cfl3d.larc.nasa.gov/>（Version 5.0 manual, TurbEq.pdf））

$$\mu_T = \frac{a_1 \rho K}{\max(a_1 \omega, f_2 \|\Omega\|)}$$

$$\frac{\partial \rho K}{\partial t} + \frac{\partial \rho v_j K}{\partial x_j} = \frac{\partial}{\partial x_j} \left[(\mu_L + \sigma_K \mu_T) \frac{\partial K}{\partial x_j} \right] + P_K - \beta^* \rho \omega K \quad (4.1)$$

$$\frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho v_j \omega}{\partial x_j} = \frac{\partial}{\partial x_j} \left[(\mu_L + \sigma_\omega \mu_T) \frac{\partial \omega}{\partial x_j} \right] + P_\omega - \beta \rho \omega^2 + 2(1 - f_1) \frac{\rho \sigma_{\omega 2}}{\omega} \frac{\partial K}{\partial x_j} \frac{\partial \omega}{\partial x_j}$$

（该方程为有量纲形式，为了书写简便，省略了物理量上方的“~”）

采用来流量作为参考量，将(4.1)方程无量纲化：

$$u = \tilde{u} / \tilde{u}_\infty, \quad \rho = \tilde{\rho} / \tilde{\rho}_\infty, \quad t = \tilde{t} \tilde{u}_\infty / \tilde{L}_\infty, \quad \mu_t = \tilde{\mu}_t / \tilde{\mu}_\infty, \quad K = \tilde{K} / \tilde{u}_\infty^2, \quad \omega = (\tilde{\mu}_\infty \tilde{\omega}) / (\tilde{\rho}_\infty \tilde{u}_\infty^2)$$

得到无量纲方程如下：

$$\mu_T = \frac{a_1 \rho K}{\max(a_1 \omega, f_2 \|\Omega\| / \text{Re})}$$

$$\frac{\partial \rho K}{\partial t} + \frac{\partial \rho v_j K}{\partial x_j} = \frac{1}{\text{Re}} \frac{\partial}{\partial x_j} \left[(\mu_L + \sigma_K \mu_T) \frac{\partial K}{\partial x_j} \right] + \frac{1}{\text{Re}} P_k - \text{Re} \beta^* \rho \omega K \quad (4.1)$$

$$\frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho v_j \omega}{\partial x_j} = \frac{1}{\text{Re}} \frac{\partial}{\partial x_j} \left[(\mu_L + \sigma_\omega \mu_T) \frac{\partial \omega}{\partial x_j} \right] + \frac{1}{\text{Re}} P_\omega - \text{Re} \beta \rho \omega^2 + \frac{1}{\text{Re}} 2(1 - f_1) \frac{\rho \sigma_{\omega 2}}{\omega} \frac{\partial K}{\partial x_j} \frac{\partial \omega}{\partial x_j}$$

其中 K 为湍动能, $\omega = \varepsilon / K$ 为湍动能的比耗散率。该模型是 $k-\varepsilon$ 模型与 $k-\omega$ 模型的混合模型。在近壁区 (边界层内), 该模型切换为 $k-\omega$ 模型; 而在远离壁面的区域, 该模型切换为 $k-\varepsilon$ 模型。这样可以兼具二者的优点。 $k-\omega$ 模型无需近壁衰减函数, 对近壁湍流粘性系数预测较好; $k-\varepsilon$ 模型对来流条件 (来流的 k, ε 值) 不敏感, 因而预测边界层外流动更为合适。

其中, 源项为:

$$P_k = \mu_T \Omega^2, \quad P_\omega = C_\omega \rho \Omega^2$$

为了实现 $k-\varepsilon$ 模型与 $k-\omega$ 模型之间的切换, SST 模型引入了识别函数 f_1 , 该函数定义为:

$$\begin{aligned} f_1 &= \tanh(\arg_1^4), \\ \arg_1 &= \min \left[\max(\Gamma_1, \Gamma_3), \Gamma_2 \right] \\ \Gamma_1 &= \frac{1}{\text{Re}^2} \frac{500 \mu_L}{\rho \omega d^2}, \Gamma_2 = \frac{4 \rho \sigma_{\omega 2} K}{CD_{K\omega} d^2}, \Gamma_3 = \frac{1}{\text{Re}} \frac{\sqrt{K}}{0.09 \omega d} \\ CD_{K\omega} &= \max \left(2 \frac{\rho \sigma_{\omega 2}}{\omega} \frac{\partial K}{\partial x_j} \frac{\partial \omega}{\partial x_j}, 10^{-20} \right) \end{aligned}$$

其中 d 为该点到(最近)壁面的距离。从上式可以看出, 随着 d 的减小, \arg_1 值快速增加, 导致 f_1 值接近 1。

模型常数 $\sigma_K, \sigma_\omega, \beta, C_\omega$ 也是 $k-\varepsilon$ 模型与 $k-\omega$ 模型中系数的组合:

$$\phi = f_1 \phi_1 + (1 - f_1) \phi_2$$

其中 ϕ_1 与 ϕ_2 分别为 $k-\omega$ 及 $k-\varepsilon$ 模型给定的系数值:

$$\begin{aligned}\sigma_{K1} &= 0.85, \sigma_{\omega1} = 0.5, \beta_1 = 0.075, C_{\omega1} = 0.533 \\ \sigma_{K2} &= 1.0, \sigma_{\omega2} = 0.856, \beta_2 = 0.0828, C_{\omega2} = 0.440\end{aligned}$$

其他模型系数为： $a_1 = 0.31, \beta^* = 0.09, \kappa = 0.41$

$$f_2 = \tanh(\arg_2^2), \quad \arg_2 = \max(2\Gamma_3, \Gamma_1)$$

f_2 也是在近壁处趋近于 1。

4.3.2 边界条件

$$\text{固壁边界: } K = 0, \omega = 10 \frac{6\mu_L}{\rho\beta_1 d_1^2} \frac{1}{\text{Re}^2}$$

$$\text{自由来流: } K_\infty = 9 \times 10^{-9}, \omega_\infty = 1 \times 10^{-6}, \mu_{T\infty} = 9 \times 10^{-3},$$

4.3.3 k 及 ω 方程数值计算过程

k, ω 方程可以统一写出如下形式：

$$\frac{\partial \varphi}{\partial t} + \frac{\partial v_j \varphi}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\tilde{\mu} \frac{\partial \varphi}{\partial x_j} \right] + Q$$

其中

$\varphi = \rho k$ 或 $\rho \omega$ ， $\tilde{\mu} = \mu_L + \sigma \mu_T$ ， Q 为源项。在 (I, J, K) 点为中心的控制体上，对上式进行积分，得：

$$\Omega_{IJK} \frac{\partial \varphi_{IJK}}{\partial t} + \oint_{\partial \Omega} (f_n - f_n^v) ds = \Omega_{IJK} Q_{IJK}$$

对流通量为 $f_n = \varphi \vec{v} \cdot \vec{n}$ ，扩散通量为 $f_n^v = \tilde{\mu} \frac{\partial \varphi}{\partial x_j} n_j$ 。对流项可以采用一阶迎风格式进行离

散（该方程无需太高精度），扩散项可以采用二阶中心格式离散。源项中的导数可以采用二阶中心差分离散。

采用一阶迎风格式，配合 L-F 分裂的对流通量为：

$$f_{j+1/2} = f_j^+ + f_{j+1}^- = \left[\frac{1}{2} (u_n + |u_n|)_j \varphi_j + \frac{1}{2} (u_n - |u_n|)_{j+1} \varphi_{j+1} \right]$$

也可采用一阶 Roe 格式计算对流通量，公式为：

$$f_{j+1/2} = \frac{1}{2}[(u_n \varphi)_j + (u_n \varphi)_{j+1}] - \frac{1}{2}[[a](\varphi_{j+1} - \varphi_j)]$$

$$a = \frac{(u_n \varphi)_{j+1} - (u_n \varphi)_j}{u_{n,j+1} - u_{n,j}}$$

4.3.4 k- ω 方程的 LU-SGS 隐式时间推进

为了增加时间步长，提高计算效率， k, ω 方程经常采用隐式求解。LU-SGS 是常用的隐式时间推进方法。此外，采用隐格式也可克服模型方程的刚性，提高计算的稳定性。

采用 Euler 隐格式对时间导数进行离散后得：

$$\Omega \frac{\varphi^{n+1} - \varphi^n}{\Delta t} + \oint_{\partial\Omega} \tilde{f}_n^{n+1} ds - \Omega Q^{n+1} = 0$$

其中 $\tilde{f}_n = f_n - f_n^v$

方程两端同时加上 $-(\oint_{\partial\Omega} \tilde{f}_n^n ds - \Omega Q^n) \triangleq R^n$ ，得：

$$\Omega \frac{\varphi^{n+1} - \varphi^n}{\Delta t} + \oint_{\partial\Omega} (\tilde{f}_n^{n+1} - \tilde{f}_n^n) ds - \Omega(Q^{n+1} - Q^n) = R^n$$

令 $w^n = \varphi^{n+1} - \varphi^n$ ，代入上式，利用

$$\tilde{f}_n^{n+1} - \tilde{f}_n^n \approx A^n w^n, \quad Q^{n+1} - Q^n \approx B^n w^n$$

其中， $A = \frac{\partial \tilde{f}}{\partial \varphi}$, $B = \frac{\partial Q}{\partial \varphi}$ 分别为通量项和源项的 Jacobian 矩阵（系数）。

就会得到关于 w^n 的方程：

$$\Omega \left(\frac{1}{\Delta t} - B^n \right) w^n + \oint_{\partial\Omega} (A^n w^n) ds = R^n \quad (4.3)$$

如源项的刚性不是很强，则在隐式部分可以不考虑其影响（相当于源项显式计算），则上式可简化为：

$$\frac{\Omega}{\Delta t} w^n + \oint_{\partial\Omega} (A^n w^n) ds = R^n$$

使用 LU-SGS 方法求解即可（见 3.2 节求解流场的 LU-SGS 方法）。实际上，两个标量方程的 LU-SGS 求解过程与求解流场的 LU-SGS 方程非常相似。在编制程序时，仅需要将守恒

变量的数目由 5 个增加到 7 个即可，LU-SGS 过程无需修改（即，最后两个标量方程的求解与前面 5 个方程相同）。

4.3.5 k, ω 方程的源项刚性的处理方法

有些情况下， k, ω 方程的源项具有刚性，因而隐式部分需要考虑源项的影响。考虑到 k, ω 方程（4.1）的复杂性，精确计算 $B = \frac{\partial Q}{\partial \phi}$ 的难度很大。作者认为，在多数计算中，网格在近壁处密集，从而制约了时间步长。因而主要的刚性来自近壁处，而在近壁处，由边界条件：

$$K = 0, \omega = 10 \frac{6\mu_L}{\rho\beta_1 d_1^2} \frac{1}{\text{Re}^2}$$

可知，当第一层网格距壁面很近时， ω 的值会非常大，从而带来了刚性。作者认为，方程（4.1）中， k 方程的 $-\beta^* \rho \omega K$ 与 ω 方程中 $-\beta \rho \omega^2$ 源项刚性最强。其 Jacobian 系数分别为 $B^K = -\beta^* \omega \text{Re}$ 及 $B^\omega = -2\beta \omega \text{Re}$ 。将其代入（4.3）式就得到 k 方程及 ω 方程的隐格式方程：

$$\Omega \left(\frac{1}{\Delta t} + \beta^* \omega^n \text{Re} \right) w^n + \oint_{\partial\Omega} (A^n w^n) ds = R^n$$

$$\Omega \left(\frac{1}{\Delta t} + 2\beta \omega^n \text{Re} \right) w^n + \oint_{\partial\Omega} (A^n w^n) ds = R^n$$

考虑源项后，LU-SGS 的计算步骤仍与 3.2.4 节介绍的相同，仅仅是在计算对角线元素（3.2.20 式）时，需要添加上 $-B^K = \beta^* \omega$ 及 $-B^\omega = 2\beta \omega$ 。即，在计算 k 方程和 ω 方程时，分别将(3.2.20 式)的 α 替换为 $\alpha + \beta^* \omega \text{Re} \Omega$ 和 $\alpha + 2\beta \omega \text{Re} \Omega$ 。

但，实际测试发现，该方法效果并不理想，因而目前版本（ver 0.98）并未采用该处理方法。即 并未将 α 替换为 $\alpha + \beta^* \omega \text{Re} \Omega$ 和 $\alpha + 2\beta \omega \text{Re} \Omega$ ；

4.3.6 计算测试

计算发现，直接采用 Blazek 书(Computational fluid dynamic: Principles and applications)中的 SST $k-\omega$ 模型公式计算出的湍流粘性系数与 BL 模型相比明显偏大。与 BL 及 SA 模型结果相比，超出了 10 倍以上。过大的湍流粘性系数掩盖了背风面头部的弱分离区，造

成压力系数的计算偏差（图 4 中红色箭头所指的区域）。采用（4.2）式对湍能方程的源项进行限制，效果会好一些。图 1-3 给出了 BL 模型、不含源项限制的 SST 模型以及包含源项限制的 SST 模型给出的湍流粘性系数。可以看出包含源项限制后，湍流粘性系数降为了原先的 1/3，粘性系数过高的现象有所改善。图 4 给出了采用不同湍流模型计算出的 RAE2822 翼型表面的压力系数，可以看出 SST 模型计算结果优于 BL 模型，特别是添加了源项限制的 SST 模型。

软件 OpenCFD-EC ver 1.4 中的 SST 模型对湍能生成项采用(4.2)式的限制器。

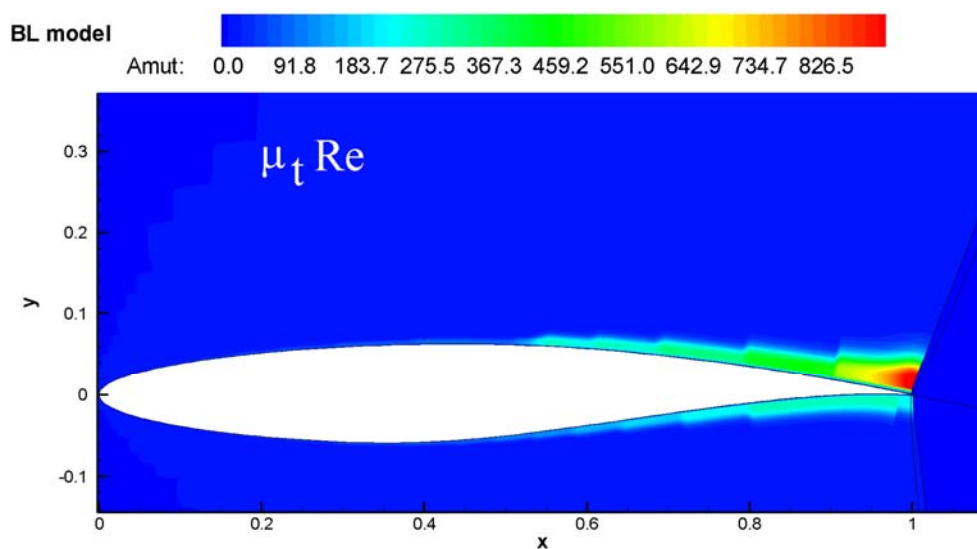


图 1 采用 BL 模型计算出的 RAE2822 周围的湍流粘性系数 ($\mu_t Re$)

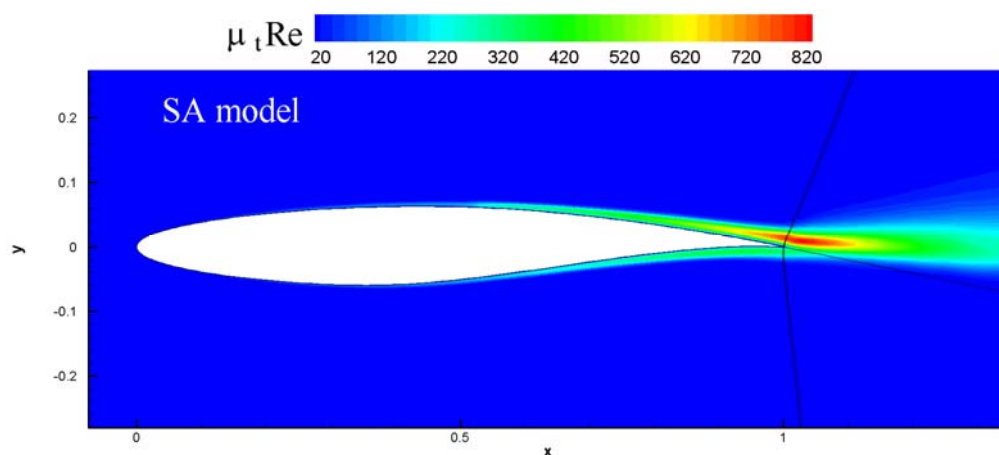


图 2 采用 SA 模型计算出的 RAE2822 周围的湍流粘性系数

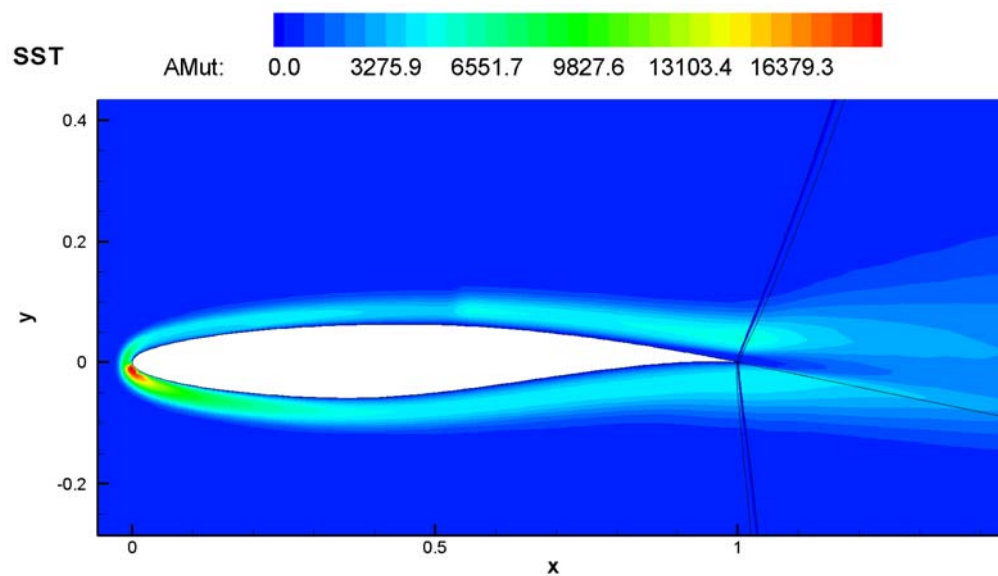


图 3 采用 SST (Blazek's Book)模型计算出的 RAE2822 周围的湍流粘性系数 ($\mu_t \text{Re}$)

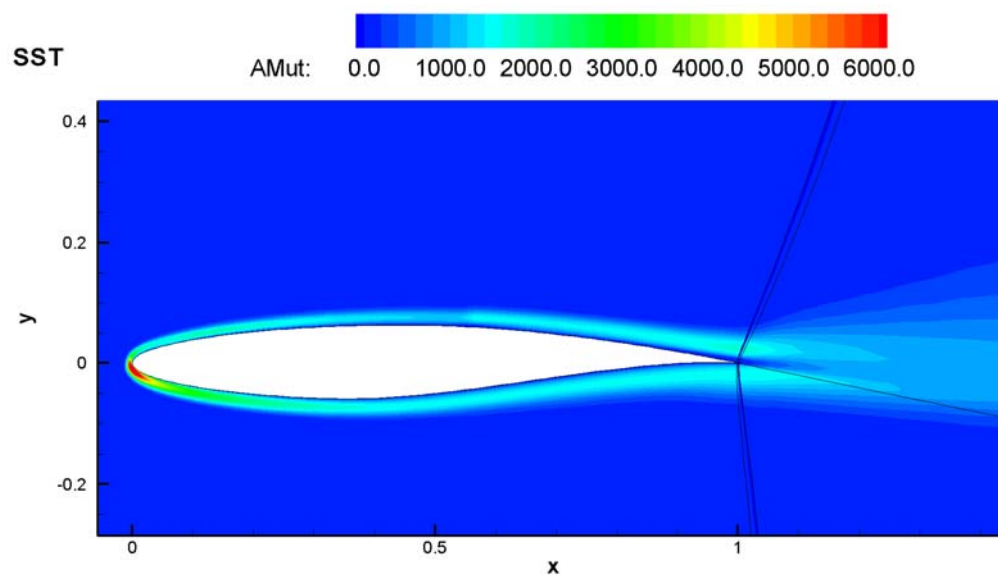


图 4 采用包含源项限制(4.2)的 SST 模型计算出的 RAE2822 周围的湍流粘性系数 ($\mu_t \text{Re}$)

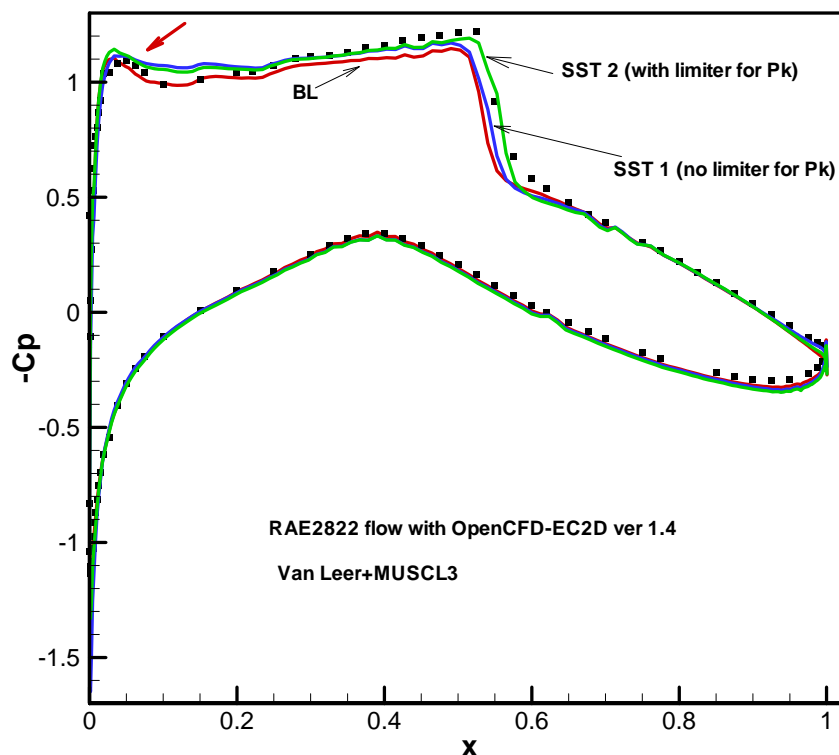


图 5 表面压力系数分布

第五章 有限差分-有限体积混合方法

OpenCFD-EC 2D 代码从 ver 1.5 版本开始，3D 代码从 ver 0.83 版本开始支持有限体积-有限差分混合计算（该功能以前是本课题组软件 OpenCFD-SEC 的功能，目前融入本软件中）。

有限体积法(FVM)与有限差分法(FDM)是 CFD 中最常使用的两种数值方法。有限体积法具有守恒性好、鲁棒性强及适应复杂几何外形的优点，常用于复杂外形工程问题的计算。但受其方法限制，高精度有限体积法构造起来困难，通常使用的有限体积法精度为二阶。有限差分法原理及实现简单，易于实现高精度计算，常用于模型问题及流动机理问题的计算。但有限差分法要求网格充分光滑，适应复杂网格的能力较差。

鉴于有限体积、有限差分方法的优缺点，本软件将有限体积-有限差分法混合使用。在网格质量较差或对局部分辨率要求不高的区域使用有限体积法；在网格质量较好，且对局部分辨率有较高要求的区域使用有限差分法。这样可发挥有限体积及有限差分方法的优点，即可实现复杂外形整体流动的正确模拟，有可实现局部区域的高分辨率模拟。

5.1 有限体积-有限差分混合方法的基本原理

有限体积-有限差分混合方法基于多块结构网格。用户可通过控制文件（本软件为 `fvm_fdm.in`）来指定各块采用何种方法（有限体积或有限差分法）来计算。

考虑到多块结构网格块-块衔接区域通常不保证网格的光滑过渡，因而即使某块采用高精度有限差分法，在靠近块边界层的几层（例如 3-4 层，视差分格式的基架点宽度而定）网格仍采用有限体积法计算。因而，“有限差分块”实际上是混合计算块。

在程序具体实现过程中，为了对原 OpenCFD-EC 改动最小化，有限差分方法模块作为“补丁”，嵌入主程序（有限体积程序，即原先版本的 OpenCFD-EC）中。具体为：主程序（有限体积程序）计算完残差后，使用有限差分方法计算块内部点的残差，并替换掉原先有限体积法计算的残差。这种实现方法对主程序改动很小，实际上，这种方法对主程序(原先版本的 OpenCFD-EC)的改动不超过 10 行代码。为了提高效率，还可对主程序进行进一步改动。例如不重复计算残差（如果某一区域使用有限差分法计算，则有限体积法部分则跳过该区域的计算）。

有限差分法仅替换了原有限体积法中计算通量的模块，而其他模块（如多重网格、隐格式、湍流模型计算）仍为有限差分法与有限体积法共用。

本程序的有限差分模块包括两个文件：`sub_comput_Jacobian.f90` 及 `sub_comput_Jacobian.f90`。前者是计算 Jacobian 变换系数，后者是利用差分方法计算残差。该模块的独立性较强，便于移植到其他结构网格有限体积程序中。

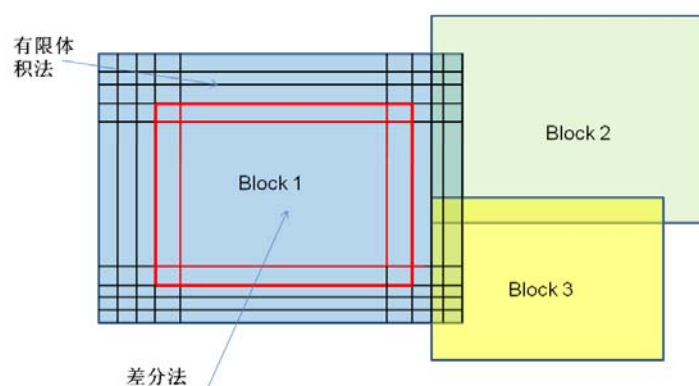


图 1 网格示意图

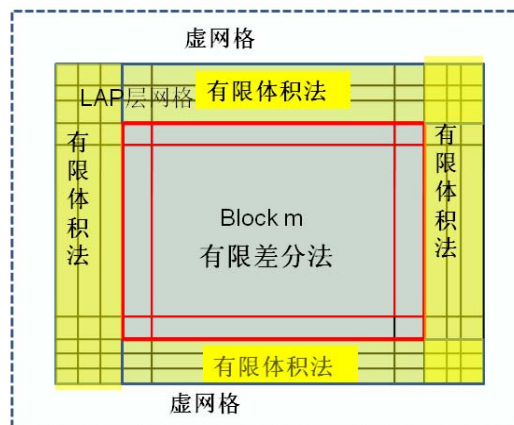


图 2 块内部有限体积-有限差分混合方法示意图

有限体积-有限差分混合方法基于多块结构网格，其实现过程见图 1 及图 2。对于其中的一块，靠近边界的 LAP 层网格（本软件的目前版本中，设定 LAP=3）采用有限体积法，内部网格采用有限差分法计算。

5.2 有限差分法模块基本原理

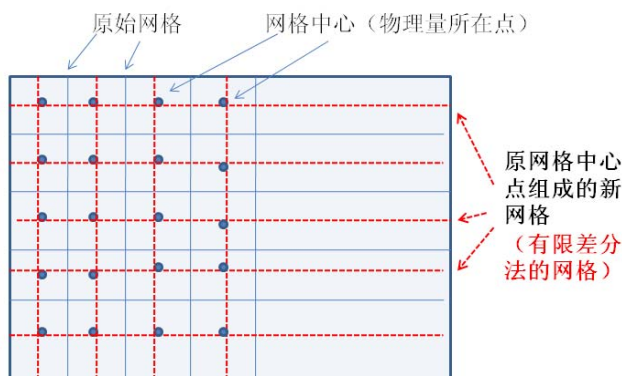


图 3 有限差分法的网格

5.2.1 有限差分法的网格系统

有限差分法计算物理量均存储于**网格点**上，而通常情况下，有限体积法物理量多存储于**网格中心**（例如，OpenCFD-EC 的物理量就存储于网格中心。当然，也有存储于网格节点上的）。为了有限差分-有限体积的网格互相匹配，可以网格中心为节点构建**第二套网格**（见图 3），有限差分法将基于该网格。

5.2.2 控制方程

有限差分法基于曲线坐标系下的 N-S 方程：

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} + \frac{\partial \hat{G}}{\partial \varsigma} = \frac{\partial \hat{E}_v}{\partial \xi} + \frac{\partial \hat{F}_v}{\partial \eta} + \frac{\partial \hat{G}_v}{\partial \varsigma} \quad (1)$$

其中：

$$\begin{aligned}
\hat{U} &= J^{-1}U \\
\hat{E} &= J^{-1}(\xi_x E + \xi_y F + \xi_z G) \\
\hat{F} &= J^{-1}(\eta_x E + \eta_y F + \eta_z G) \\
\hat{G} &= J^{-1}(\zeta_x E + \zeta_y F + \zeta_z G) \\
\hat{E}_v &= J^{-1}(\xi_x E_v + \xi_y F_v + \xi_z G_v) \\
\hat{F}_v &= J^{-1}(\eta_x E_v + \eta_y F_v + \eta_z G_v) \\
\hat{G}_v &= J^{-1}(\zeta_x E_v + \zeta_y F_v + \zeta_z G_v)
\end{aligned}$$

$U = [\rho, \rho u, \rho v, \rho w, E]^T$ 是 N-S 方程的守恒变量, E, F, G 为三个方向的无粘通量, E_v, F_v, G_v 是三个方向的粘性通量。

Jacobian 变换系数的定义为:

$$J = \frac{\left| \frac{\partial(\xi, \eta, \zeta)}{\partial(x, y, z)} \right|}{\left| \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} \right|}^{-1},$$

$$\begin{aligned}
\xi_x &= J(y_\eta z_\zeta - z_\eta y_\zeta) \\
\xi_y &= J(z_\eta x_\zeta - x_\eta z_\zeta) \\
\xi_z &= J(x_\eta y_\zeta - y_\eta x_\zeta) \\
\eta_x &= J(y_\zeta z_\xi - z_\zeta y_\xi) \\
\eta_y &= J(z_\zeta x_\xi - x_\zeta z_\xi) \\
\eta_z &= J(x_\zeta y_\xi - y_\zeta x_\xi) \\
\zeta_x &= J(y_\xi z_\eta - z_\xi y_\eta) \\
\zeta_y &= J(z_\xi x_\eta - x_\xi z_\eta) \\
\zeta_z &= J(x_\xi y_\eta - y_\xi x_\eta)
\end{aligned}$$

在程序中, 这些系数采用差分法计算, 计算域内部采用 6 阶中心差分格式计算, 临近边界的地方采用 4 阶及 2 阶差分格式计算。

5.2.3 无粘项的计算

1) 通量分裂技术

目前版本支持基于流通矢量分裂(FVS)的差分方法。 后续版本还将支持基于通量差分分裂(FDS)及 AUSM 类通量分裂的差分方法。

采用 Steger-Warming 通量分裂技术, 将通量分裂成正、负通量:

$$\hat{E} = \hat{E}^+ + \hat{E}^-, \quad \hat{F} = \hat{F}^+ + \hat{F}^-, \quad \hat{G} = \hat{G}^+ + \hat{G}^-$$

具体公式如下:

设: $f = \alpha_1 E + \alpha_2 F + \alpha_3 G$

其中 E, F, G 为 3 个方向的流通矢量。其特 Jacobian 矩阵的征值为：

$$\lambda_{1,2,3}(A) = \tilde{V} \quad \lambda_{4,5} = \tilde{V} \mp c\sigma$$

$$\text{其中 } \tilde{V} = \alpha_1 u + \alpha_2 v + \alpha_3 w \quad \sigma = \sqrt{\alpha_1^2 + \alpha_2^2 + \alpha_3^2}$$

将征值分解为正特征值和负特征值两部分：

$$\lambda_k = \lambda_k^+ + \lambda_k^- \quad (k=1,2,3,4,5), \quad \text{其中: } \lambda_k^+ = \frac{\lambda_k + |\lambda_k|}{2}, \quad \lambda_k^- = \frac{\lambda_k - |\lambda_k|}{2}$$

用 λ_k^+ 和 λ_k^- 替代下式中的 $\tilde{\lambda}_k$ 便可得到三维流通矢量 f 的正、负通量 f^\pm ：

$$f = \frac{\rho}{2\gamma} \begin{bmatrix} 2(\gamma-1)\tilde{\lambda}_1 + \tilde{\lambda}_4 + \tilde{\lambda}_5 \\ 2(\gamma-1)\tilde{\lambda}_1 u + \tilde{\lambda}_4(u - c\tilde{k}_1) + \tilde{\lambda}_5(u + c\tilde{k}_1) \\ 2(\gamma-1)\tilde{\lambda}_1 v + \tilde{\lambda}_4(v - c\tilde{k}_2) + \tilde{\lambda}_5(v + c\tilde{k}_2) \\ 2(\gamma-1)\tilde{\lambda}_1 w + \tilde{\lambda}_4(w - c\tilde{k}_3) + \tilde{\lambda}_5(w + c\tilde{k}_3) \\ (\gamma-1)\tilde{\lambda}_1 V^2 + \frac{\tilde{\lambda}_4}{2}[(u - c\tilde{k}_1)^2 + (v - c\tilde{k}_2)^2 + (w - c\tilde{k}_3)^2] \\ + W_{II} + \frac{\tilde{\lambda}_5}{2}[(u + c\tilde{k}_1)^2 + (v + c\tilde{k}_2)^2 + (w + c\tilde{k}_3)^2] \end{bmatrix} \quad (2)$$

其中：

$$\tilde{k}_1 = \frac{\alpha_1}{\sigma}, \quad \tilde{k}_2 = \frac{\alpha_2}{\sigma}, \quad \tilde{k}_3 = \frac{\alpha_3}{\sigma}, \quad \sigma = \sqrt{\alpha_1^2 + \alpha_2^2 + \alpha_3^2}$$

$$W_{II} = \frac{(3-\gamma)(\tilde{\lambda}_4 + \tilde{\lambda}_5)c^2}{2(\gamma-1)}$$

分别代入 (2) 得到正通量和负通量：

$$f = f^+ + f^-$$

具体公式见 《计算空气动力学》（傅德薰等编著）160 页。

2) 差分格式

目前版本使用 5 或 7 阶 WENO 格式计算无粘通量。今后版本还将补充更高精度及分辨率的差分格式。

正通量的 5 阶 WENO 格式离散的形式为：

$$F_{j+1/2}^{WENO5} = \omega_1 q_1 + \omega_2 q_2 + \omega_3 q_3 \quad (3)$$

$$\omega_k = \frac{\alpha_k}{\alpha_1 + \alpha_2 + \alpha_3}, \quad \alpha_k = \frac{C_k}{(\varepsilon + IS_k)^2}, \quad \varepsilon = 10^{-6}$$

$$C_1 = 1/10, C_2 = 6/10, C_3 = 3/10$$

$$q_1 = (2f_{j-2} - 14f_{j-1} + 11f_j)/6 \quad q_2 = (-f_{j-1} + 5f_j + 2f_{j+1})/6 \quad q_3 = (2f_j + 5f_{j+1} - f_{j+2})/6$$

$$IS_1 = \frac{13}{12}(f_{j-2} - 2f_{j-1} + f_j)^2 + \frac{1}{4}(f_{j-2} - 4f_{j-1} + 3f_j)^2$$

$$IS_1 = \frac{13}{12}(f_{j-1} - 2f_j + f_{j+1})^2 + \frac{1}{4}(f_{j-1} - f_{j+1})^2$$

$$IS_1 = \frac{13}{12}(f_j - 2f_{j+1} + f_{j+2})^2 + \frac{1}{4}(f_j - 4f_{j+1} + 3f_{j+2})^2$$

以上为正通量的 5 阶 WENO 差分格式。对于负通量，将上式中下标的“j+k”替换为“j-k”即可（例如把“j+2”替换为“j-2”，把“j-1”替换为“j+1”），然后将(3.2.2)式的“j+1/2”替换为“j-1/2”就可以了。

5.2.4 粘性项的计算方法

当前版本差分模块中粘性项一律用四阶精度中心差分格式计算。

第六章 OpenCFD-EC 的数据结构及流程

§6.1 二维程序 (OpenCFD-EC2D ver 1.1)

6.1.1 数据结构

1. 全局数据 **Global_Var**

定义在 **module Global_Var** 模块中。程序中的多数子程序通过 Use Global_Var 语句使用了该模块, 因此该模块中的数据属于**全局数据**。包括计算参数、几何量及物理量等。

1) 全局定义的参数

双精度数: Ma, Re, gamma, Pr, AoA, t_end, Cp, Cv, p00。含义分别为: Mach 数, Reynolds 数, 比热比(1.4), Prandtl 数, 攻角, 计算结束的时刻, 定压比热, 定容比热, $p00 = \frac{1}{\gamma Ma^2}$ 。

整形参数:

Istep_save, Iflag_turbulence_model, Iflag_init, Iflag_Scheme, IFlag_flux, Iflag_local_dt, IFlag_Reconstruction, Time_Method

这些量是计算的控制变量。

Istep_save: 控制数据存盘的频率(Istep_save 步存一次盘)

Iflag_turbulence_model: 湍流模型。0 不使用湍流模型, 1 BL 模型

Iflag_init: 初始化方法。0 利用来流初始化, 1 利用数据文件 flow2d.dat 初始化。

Iflag_scheme: 数值格式; (Scheme_NND2=1, Scheme_UD3=2, Scheme_WENO3=3, Scheme_MUSCL2=4, Scheme_MUSCL3=5)

Iflag_flux: 通量方法; (Flux_Steger_Warming=1, Flux_HLL=2, Flux_HLLC=3, Flux_Roe=4, Flux_VanLeer=5, Flux_Ausm=6)

Iflag_local_dt: 0 不采用局部时间步长; 1 采用局部时间步长法。

IFlag_Reconstruction: 通量重构方式; 0 利用原始变量、1 利用守恒变量、2 利用特征变量

Time_Method: 时间推进方法; 1 1 阶 Euler; 3 3 阶 Runge-Kutta

2) 数据结构 **BC_MSG_TYPE**, 描述**边界连接信息**, 定义为:

TYPE BC_MSG_TYPE

integer:: f_no, face, ist, iend, jst, jend, neighb, subface, orient

END TYPE BC_MSG_TYPE

该数据结构描述一个“子面”。包含 10 个整形数据变量，描述一个子面(subface)。

含义依次为：面号(face number), 面的类型(1-4 分别表示 i-, j-, i+,j+面), i 方向的起点下标, i 方向的终点下标, j 方向的起点下标, j 方向的终点下标, 相邻的块号, 相邻的子面号, 连接指向。

“子面”等概念见附录中的《多块结构网络连接信息搜索算法》。

3) 数据结构 **Block_TYPE** , 描述一个“块”

所谓的“块”是多块结构网格中的一个连续的区域, 块内的网格为(单块)结构网格。

其定义为:

```
TYPE Block_TYPE
integer :: Block_no,nx,ny,subface      ! 块号; 网格数 nx,ny; 子面数
real*8,pointer,dimension(:,,:,:) :: x,y,x1,y1  ! 网格节点坐标; 网格中心坐标
real*8,pointer,dimension(:,,:,:) :: s0        ! 控制体面积
real*8,pointer,dimension(:,,:,:) :: U,Un      ! 守恒变量 (本时间步及上一个时间步的值)
real*8,pointer,dimension(:,,:,:) :: Res       ! 残差 (净通量)
real*8,pointer,dimension(:,,:,:) :: dt        ! (局部)时间步长
real*8,pointer,dimension(:,,:,:) :: QF        ! 强迫函数 (多重网格法中粗网格使用)
real*8,pointer,dimension(:,,:,:) :: deltU     ! 守恒变量的差值, dU=U(n+1)-U(n) 多重网格使用
TYPE(BC_MSG_TYPE),pointer,dimension(:,,:,:) :: bc_msg ! 边界链接信息
End TYPE Block_TYPE
```

其中包含了指向几何量及物理量的指针。这些指针将在初始化模块中开辟内存空间。

该数据结构保存了计算中必要的几何量和物理量。几何量包括网格节点坐标、网格中心点坐标及控制体(即网格)的面积。物理量包括定义在网格中心点上的守恒变量(质量密度、动量密度和能量密度)。为了使用多重网格法, 残差也被保存下来, 残差为积分型 N-S 方程的右端项, 即单位时间内进入控制体的净通量。

边界连接信息是“块”的属性。因此 bc_msg (BC_MSG_TYPE 结构)也是 Block_TYPE 的一个数据成员。

4) 数据结构 **Mesh_TYPE**, 描述一套“网格”

所谓“网格”是由全部“块”组成的集合, 其数据成员包括几何量及物理量。如果不开启多重网格机制, 只使用一套网格。开启多重网格的情况下, 可使用多套网格, 包括细网格、粗网格、最粗网格。本软件目前最多支持三重网格。如使用更多重网格, 可在原代码基础上进行微量修改。

其定义为:

```
TYPE Mesh_TYPE
integer :: Mesh_no,Num_Block, Num_Cell ! 网格编号 (1 号为最细网格, 2 号为粗网格, 3 号为更
```

```

粗网格...), 网格块数, 网格数目
real*8:: Res_max(4),Res_rms(4)                ! 最大残差, 均方根残差
TYPE (Block_TYPE),pointer,dimension(:):: Block    ! “块”    (从属于“网格”)
!        从属于网格的控制信息
integer:: Iflag_turbulence_model, Iflag_Scheme,IFlag_flux,IFlag_Reconstruction
End TYPE Mesh_TYPE

```

可以看出,“网格”的主要数据成员是一个“块”的数组。除此之外,还有网格编号、包含块的数目及网格单元数等数据成员。网格编号 1 表示最细网格,2 表示粗网格,3 表示更粗网格。

数据结构“网格”里面还包含了一些整形的控制变量: Iflag_turbulence_model, Iflag_Scheme,IFlag_flux,IFlag_Reconstruction, 控制该网格所使用的湍流模型、数值格式、通量技术及重构方式。这些变量从属于“网格”,因此不同网格可以采用不同的数值方法及湍流模型。

一般情况下,最细网格采用控制文件 **control.in** 指定的数值方法及湍流模型,而粗网格采用低精度的数值方法(如 1 阶迎风)且不采用湍流模型。

5) 全局可见的临时变量

以指针的形式定义为:

```

real*8, save,pointer,dimension(:,:,:):: d,uu,v,T,p,cc, Amu
real*8, save,pointer,dimension(:,:,:):: Fluxi,Fluxj

```

其物理含义为:

d: 密度; uu,v: x-及 y-方向的速度; T: 温度; p: 压力; cc: 声速;

Amu: (总)粘性系数。以上量均定义在**网格中心点**。

Fluxi 及 Fluxj 为 i-及 j-方向控制体(网格)界面的流通量。

以上物理量并不从属于某套“网格”(Mesh) 或某个“块”(Block), 属于**临时变量**。

在计算过程中,进入某个块进行计算时,通过 **allocate()** 给这些变量**临时开辟内存**,退出该块计算时,通过 **deallocate()** **收回**这些内存。

这样可以节省内存。

6.1.2 计算流程

- 1) call read_parameter 读取流动参数及控制信息
- 2) call check_mesh_multigrid 检查网格配置所允许的最大重数,并设定多重网格的重数。例如每块网格每个维上的点数是 $2*n+1$ 时,允许设定为 2 重网格;点数为 $4*n+1$ 时,允许设定为 3 重网格 (本版本最多可设定为 3 重网格)。计算中使用的网格重数由控制文件 control.in 中的 Num_Mesh 值给定;如果该值超出了网格所允许的值,则程序会按照较小的重数进行计算。
- 3) call Init 初始化,创建数据结构;读入几何及物理信息。
 - a. 创建数组 Mesh: allocate(Mesh(Num_Mesh)); 该数组是 Mesh_TYPE 类型,包含了一套网格所拥有的数据结构 (见前一节)。
 - b. Creat_Mesh1 创建主网格 Mesh(1) ; 该过程会给该网格所拥有的数据结构分配内存空间。包括几何量及物理量。其中内存占用最多的是“网格块”(Block) 指针所指向的空间。
 - c. 读入边界连接信息 call read_bcinfo; 从边界连接文件 bc2d.in 读入信息,储存在 Mesh(1)%Block(:)%bc_msg(:) 里。
 - d. 根据网格 1 (主网格) 信息,创建网格 2 (粗网格)
 - e. 根据网格 2 信息,创建网格 3 (更粗网格,三重网格时使用)。创建网格时,数据结构“网格”所拥有的全部成员变量及指针均被分配内存地址;且被初始化(多数变量以 0 初始化)
- 3) call set_control_para , 设定各重网格上的控制变量。各重网格上可以使用不同的数值方法、通量分裂形式、通量重构方式及湍流模型。

本程序设定粗网格上使用 1 阶迎风格式;通量分裂及重构方式与主网格相同;粗网格上不使用湍流模型。
- 4) call Update_coordinate_buffer 利用连接信息,给出虚网格上的几何量。

“虚网格”是每个网格块向外延伸出去的区域。网格在边界上使用 1 层或 2 层虚网格,以便与其他块交换信息。同时本程序也利用虚网格上的物理量来定义边界条件。
- 5) 计算初值

如果控制文件 control.in 中的 iflag_init 为 0,则从零流场开始计算:
首先从最粗网格开始计算,初值为均匀的无穷远流;
计算到一定步数后,插值到细网格计算;
计算到一定步数后,插值到最细的网格。

如果 iflag_init 为 1,则主网格上的物理量从 flow2d.dat 文件读取 (tecplot 格式)。
6. 进行时间推进

推进方式可选择 1 阶 Euler 或 3 阶 Runge-Kutta 方法;程序支持单重、2 重及 3 重网格。

6.2 程序的并行实现

目前版本(ver 0.71 以上) 支持 OpenMP 并行。

编译方法，如使用 Intel Fortran 编译，请在编译命令行添加 `-openmp` (Linux 系统) 或 `/Qopenmp` (Windows 系统)。

Visual studio 环境下 项目-> xxx 项目的属性-> Fortran-> Command line -> 对话框中添加 `/Qopenmp`

程序运行时，需设置 `OMP_NUM_THREADS` 环境变量，例如
Set `OMP_NUM_THREADS=8`，则可使用 8 个 CPU 核心并行运行。

第七章 并行计算

实际计算中, 计算网格数目会随着问题复杂程度的增长而快速增加。对于复杂问题, 单 CPU (单线程) 计算需要消耗很长时间, 甚至无法进行。采用并行计算是十分必要的。OpenCFD-EC 支持 OpenMP 及 MPI 两种并行方式。OpenMP 是线程级并行, 易于编程实现, 但纯 OpenMP 并行只能针对共享内存的计算机, 并行规模并不能很大。而 MPI 基于区域分解算法及网络信息交换技术, 可支持大规模 (几万至几十万 CPU 核心) 的并行作业。但 MPI 程序编制较为复杂, 尤其是针对多块结构网格, 其负载均衡与通信均衡不易掌控。

作者开发的高精度差分软件 OpenCFD-SC 使用 MPI 实现了十万 CPU 核心级的高效并行运行。OpenCFD-SC 基于单块网格, 易于实现负载及通信均衡。而多块网格则较为复杂。Ver 0.85 之前的版本只支持 OpenMP 并行, 从 0.9 版开始支持 MPI 及 MPI-OpenMP 混合并行。

§7.1 OpenMP 并行

1) 基本的 OpenMP 语句

OpenMP 属线程级并行, 通过在程序中添加有关语句来实现并行的。Fortran 语言中与 OpenMP 有关的语句以关键字 “!\$” 以及 “!\$OMP” 开头。在不支持 OpenMP 时, 这些语句被当成注释处理 (“!” 在 Fortran90 里面是注释符)。

使用 Intel Fortran 编译器进行编译时, 需在编译命令中添加 “-openmp” 命令。例如:

```
ifort -O3 -o opencfd-ec2d-1.4.3-rmp.out opencfd-ec2d-1.4.3.out -openmp
```

如下是一段基本的 OpenMP 程序:

```
Print*, “aaa”  
!$OMP PARALLEL  
Print*, “bbb”  
!$OMP END PARALLEL  
Print*, “ccc”  
End
```

其中 “!\$OMP PARALLEL” 与 “!\$OMP END PARALLEL” 之间的语句是一个 OpenMP 并行域。程序执行过程如图 7.1 所示, 当运行到并行体时, OpenMP 创建 N 个线程同时执行, 而退出并行体时, 结束这些线程而只保留一个主线程。线程数可以由环境变量 OMP_NUM_THREADS 定义, 也可在程序中由函数 `omp_set_num_threads()` 设定。OpenCFD-EC 通过该函数设定线程数为 NUM_THREADS 个 (通过控制文件 control.in 指定)。

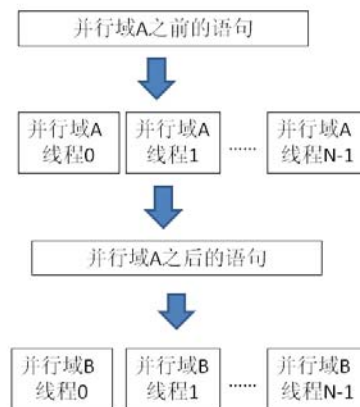


图 7.1 OpenMP 运行示意图

如果指定线程数为 4，以上代码运行后的结果为：

```
aaa
bbb
bbb
bbb
bbb
ccc
```

2) 循环并行语句

“!\$OMP DO” 与 “!\$OMP ENDDO” 指示将 Do 循环进行并行处理。 例如如下代码：

Dimension A(12),B(12)

.....

```
!$OMP PARALLEL
```

```
!$OMP DO
```

```
Do k=1,12
```

```
  A(k)=2.0*B(k)
```

```
enddo
```

```
!$OMP ENDDO
```

```
!$OMP END PARALLEL
```

假设由 4 个线程执行，则线程 0 执行 Do k=1,3; 线程 1 执行 Do k=4,6; 线程 2 执行 Do k=7,9; 线程 3 执行 Do k=10,12. 这样代码就实现了并行化。

循环并行是 OpenMP 最易于实现也是最常用的并行方式。Opencfd-EC 仅使用了这种并行方法。

需要注意的是：

- a. !\$OMP DO 需在!\$OMP PARALLE 并行域内才有效。

```
!$OMP PARALLEL
```

```
!$OMP DO
```

可以合并成 !\$OMP PARALLEL DO

同样，也有!\$OMP END PARALLE DO

b. !\$OMP DO 仅对**紧邻其后**的循环做并行展开。

例如如下代码

```
!$OMP DO
  Do k=1,Kx
    Do j=1,Jx
      A(j,k)=2.0*B(j,k)+3.0
    Enddo
  Enddo
!$OMP DO
  仅对 k 循环做了并行展开。
```

3) OpenMP 的数据域

正确使用 **OpenMP** 编程最关键是使用**正确的数据域**。OpenMP 将变量分为“共享 (SHARED)”及“私有(PRIVATE)”两大类。SHARED 变量为全部线程共享（全部线程都拥有对该变量的读-写权）。而 PRIVATE 变量属于线程私有，仅本线程才能访问。

如下段代码

```
Real: A(12),B(12), x,y
Integer:: k
x=3.0
y=2.0
.....
!$OMP PARALLEL DO
Do k=1,12
B(k)=A(k)*x+y
Enddo
!$OMP END PARALLE DO
```

显然，在并行域中，变量 A,B,x,y 均为共享变量，全部进程都能访问。而循环变量 k 则为私有变量。因为线程 0 执行 do k=1,3; 线程 1 执行 do k=4,6; 线程 2 执行 do k=7,9; 线程 3 执行 do k=10,12 （假设 4 个线程执行）。显然每个线程中的 k 为不同的数，因而该变量为私有变量。

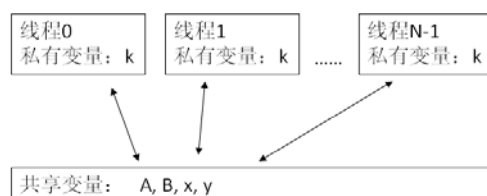


图 7.2 OpenMP 变量的作用域

OpenMP 基于共享存储，因而大部分变量默认为共享变量。循环变量默认为私有变量。

为了能够得到正确的结果，需要指定变量的共享及私有属性。格式为：

!\$OMP PARALLEL SHARED (共享变量列表....) PRIVATE (私有变量列表.....)

也可用关键字 **DEFAULT** 指定默认为共享或私有，例如
!\$OMP PARALLEL DEFAULT (SHARED) PRIVATE (私有变量列表.....).

正确设定变量的属性对于 OpenMP 十分关键。

例如：欲计算 $b(k)=2.0*a(k)*a(k)+5.0$ ($k=1,2...12$)

代码如下：

.....

```
do k=1,12
```

```
tmp=a(k)*a(k)
```

```
b(k)=2.0*tmp+5.0
```

```
enddo
```

这里引入了临时变量 **tmp** (程序中为了代码简化，会大量使用这种变量)。应当注意，如果在并行域中，该变量应当是私有的。

则 OpenMP 代码如下：

```
!$OMP PARALLEL DO SHARED(a,b) PRIVATE(k,tmp)
```

```
do k=1,12
```

```
tmp=a(k)*a(k)
```

```
b(k)=2.0*tmp+5.0
```

```
enddo
```

```
!$OMP END PARALLEL DO
```

因为不同线程执行该并行域时，**k** 是不同的，因而计算出的 **tmp** 也是不同的。如果 **tmp** 设定为共享变量，所有的进程都操作同一个变量，则结果可能出现混乱。

常见问题：在列表中漏掉了 SHARED 变量。OpenCFD-EC 经常使用如下形式定义并行域：

```
!$OMP PARALLEL DEFAULT (PRIVATE) SHARED (list .....
```

通常，控制变量，流动参数等公共变量（例如 **Re, Ma, gama, IFLAG_Scheme**）属于共享变量。如果在 **SHARED()** 中的列表中漏掉了该变量，则这些变量会被当成线程的私有变量。其值可能是随机数，造成计算出现问题。例如列表中漏掉了 **Mach** 数 **Ma**，则在并行域中 **Ma** 可能会是 0 或者随机数，造成计算出现问题。

可以改为如下定义：

```
!$OMP PARALLEL DEFAULT (FIRSTPRIVATE) SHARED (list .....
```

这样，线程的私有变量会以主线程的变量初始化，即使在 **SHARED()** 列表中漏掉了某些公共变量，有时也不会出现问题。例如 **Mach** 数 **Ma** 应当是共享变量，但编程时没有将其添加到 **SHARED()** 中的列表。但在并行域中，**Ma** 仍会以原先（主线程）的值进行初始化，不易出现问题。

4) OpenMP 的规约操作

规约操作（求和，求积，求最大、最小值等）是常用的运算。如下段代码进行求和

```
Real:: A(N),sum
```

```
.....
```

```
Sum=0
```

```

Do k=1,N
Sum=Sum+A(k)
Enddo

```

如简单采用如下代码：

```

Real:: A(12),sum
.....
Sum=0
!$OMP PARALLEL DO SHARED(A,Sum) PRIVATE(k)
Do k=1,12
Sum=Sum+A(k)
Enddo
!$OMP END PARALLEL DO

```

则 **Sum** 为每个线程共享，所有线程都操作内存中的同一个变量。则显然这些线程必须得串行执行（否则就会有读-写冲突）。则这段代码无法并行执行。

OpenMP 提供了 **REDUCTION** 关键字来实现规约操作。

语法为 **REDUCTION** (操作符: 变量列表)，其中操作符为 +, -, *, /, MAX, MIN 等。

上例的 OpenMP 代码如下：

```

Real:: A(12),sum
.....
Sum=0
!$OMP PARALLEL DO SHARED(A) PRIVATE(k) REDUCTION(+: Sum)
Do k=1,12
Sum=Sum+A(k)
Enddo
!$OMP END PARALLEL DO

```

执行后，执行到并行域时，每个线程创建一个私有变量 **Sum**（与并行域前的 **Sum** 变量不同），然后计算每个线程计算自己的“局部和”。（例如线程 0 计算 $\text{Sum}=\text{A}(1)+\text{A}(2)+\text{A}(3)$ ；线程 1 计算 $\text{Sum}=\text{A}(4)+\text{A}(5)+\text{A}(6)$;.....）。由于每个线程有自己的私有 **Sum** 变量，线程之间没有变量读-写冲突，因而可以顺利地并行执行。退出循环体后，OpenMP 在根据规约规则，计算出主线程的 **Sum**(实际上是将各线程私有的 **Sum** 相加)。

§7.2 MPI 并行

与 OpenMP 不同，MPI 基于分布式存储概念（而 OpenMP 基于共享存储），因而其编程思想与 OpenMP 有很大区别。MPI 的运行机制是：程序启动时，系统将程序拷贝 **N**

份，作为 N 个独立的进程运行（进程数 N 在提交作业时由用户决定）。各进程拥有独立的数据空间，无法直接读写其他进程的数据。每个进程可通过 `MPI_COMM_RANK()` 函数调用来获得自己的编号，并根据编号不同来执行不同的任务或处理不同的数据。进程之间可通过调用 `MPI_SEND()` 及 `MPI_RECV()` 函数来交换数据。

本程序基于多块结构网格，“进程”与“块”的关系如图 7.3 所示。每个进程可以包含多个块。目前版本的程序不允许有跨越进程的块（图 7.3A）。允许块跨越进程（图 7.3B）可提高程序的负载均衡性，从而可实现更好的并行效率。但该方式编程更为复杂，对算法也有额外要求（例如 LU-SGS 就不适用于该方式，必须换成红-黑迭代等适用于并行分割的迭代方法）。本软件将在后续版本中支持图 7.3B 的分割方法。

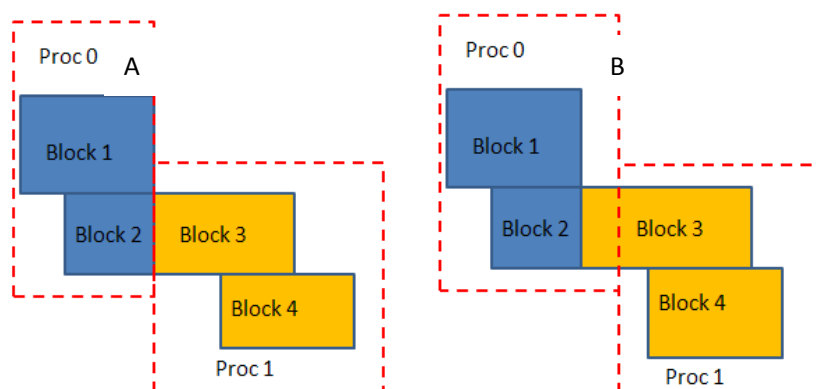


图 7.3 进程与块 (A 不允许块跨越进程， B 允许块跨越进程)

采用图 7.3A 的分割方法给程序的 MPI 并行化带来了很多便利。OpenCFD-EC 的程序核心基于每个块的单块网格有限体积法，使用外围程序实现块-块之间的信息交换及前、后处理。在并行化过程中，作为核心程序的单块网格有限体积法可不必做任何修改，仅修改作为外围程序的块-块通信模块以及 IO 等前、后处理模块。

7.2.1 块的划分与负载均衡

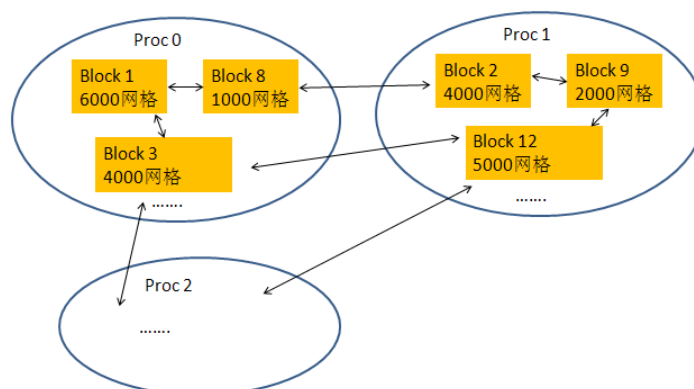


图 7.4 块的划分

如图 7.4 所示，将 Block 1, Block 2,等多个块划分给 N 个进程（通常 1 个 CPU 处理 1 个进程）。该划分需要注意负载均衡，即各进程的计算量及通信量要尽量均衡。进程内部各块之间的通信（边界数据交换）属于“本地通信”（可通过直接内存访问实现），速度很快。而跨越节点的通信只能通过 MPI_SEND()及 MPI_RECV()函数来实现，通常要通过跨节点的网络数据传输，因而速度较慢，且有一定延迟。因而，要实现较高的并行效率，在并行划分时须注意：

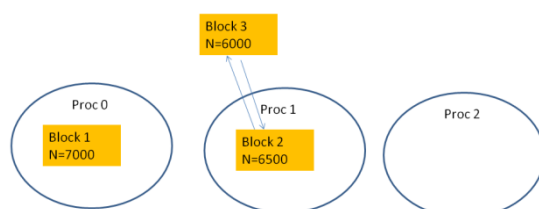
- 1) 各进程的计算量尽量均衡，即每个进程包含的总网格数基本均衡。
- 2) 尽量使用“本地通信”，即把相互连接的块放到同一个进程里。
- 3) 各进程的跨进程通信量尽量均衡。

当进程数及网格块数很多时，实现以上三点难度非常大，计算分割方案的算法也很复杂。因而多块结构网格大规模并行计算效率提高较为困难。相对而言，非结构网格由于没有“块”的概念，负载均衡反而很容易做到，因而并行效率容易提高。

[分块算法] 作者开发了区域分割算法，该算法可在不切分已有块的情况下使负载尽量均衡。算法如下：

- 1) 按各块单元（格心）点的数目进行排序，从多到少一次为 block 1, block 2,
- 2) 依次将各 block 放入进程分区(图 7.5)。原则：a.尽量先挑与本块有连接的分区；b.如没有连接，则挑选负载最小的分区；c.如加入该分区后，分区内的网格单元数超过了平均值（总网格数/总进程数）M0，则不能放入该分区。

例如，先将 block1 放入 proc0 分区。投放 block2 时，先判断其与 proc0 是否有通信关系。如没有通信，则放入 Proc1，同时判断 block2 加入后该分区的网格单元数是否超过了平均值 M0，如超过该值，则放入其他分区。



第八章 网格文件及格式转换

§8.1 Gridgen 的 Generic 格式

OpenCFD-EC3D 默认使用 Gridgen 的 Generic 格式作为网格输入格式。输入文件包含网格文件 Mesh3d.dat 以及边界描述文件 bc3d.inp.

其中网格文件 Mesh3d.dat 为 PLOT3D 格式;

bc3d.inp 为 Gridgen 的 Generic 格式;

图 1 为 Gridgen 的边界条件代码(Generic 格式)。从中可以看出,该格式的边界条件代码比较少,缺乏具体的区分。

Generic ASW Boundary Conditions

value	description
0	none
1	interblock connection
2	solid surface
3	symmetry
4	farfield
5	inflow
6	outflow
71	i pole
72	j pole
73	k pole
8	generic #1
9	generic #2
10	generic #3

图 1 Gridgen 的 Generic 边界条件代码

§8.1 CGNS 格式

CGNS (CFD General Notational System)是波音公司和 NASA 联合开发的一套 CFD 通用数据格式,目前已被广泛应用。目前大多数 CFD 软件已支持该数据格式。为了提高兼容性 & 通用性,从 0.96 版本开始,本软件的三维代码支持 CGNS 格式的网格文件。

对 CGNS 网格文件的支持主要是由前处理程序 Convert-cgns.f90 实现,该程序读入 cgns 格式的文件并转化为 Gridgen generic 格式。

8.1.1 CGNS 格式简述

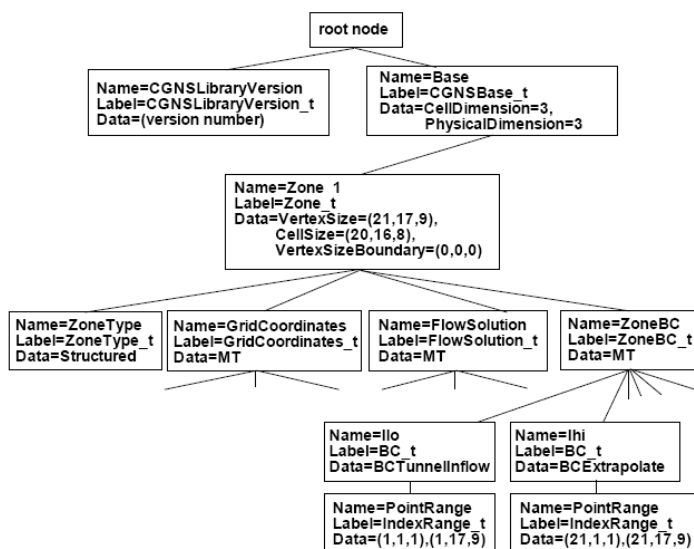


Figure 11: Layout of CGNS file for simple Cartesian structured grid with flow solution and boundary conditions using PointRange.

CGNS 采用**树型结构**存放文件。上图给出了一个典型的 CGNS 文件的结构图。

- 根节点下包含了版本信息以及一个或多个“库”（base）。对于本软件，仅需要一个库(base)。
- 每个库（base）包含了若干区(Zone)，每个区对应本程序多块网格中的一个网格块。
- 每个区包含若干数据域；这些数据域包括：
 - 坐标数据；
 - 边界区域描述（包括物理边界条件及内边界信息）；
 - 流场数据（本软件不需要）；
 - 无量纲参考量值；
 -

由于 CGNS 采用了树型结构，因而可以存放各种不同类型的数据信息。

8.1.2 CGNS 工具库的安装及数据文件的读写

CGNS 数据文件的读写通常需要调用 CGNS 库函数来实现。如需使用 CGNS 库函数，首先需要**安装CGNS 套件**。具体方法为：

- 1) 在 CGNS 官网 (<http://cgns.sourceforge.net/>) 下载 CGNS 套件(如 cgnslib_3.1.4.tar.gz) 到本地，解压；
- 2) 安装该套件。在 linux 系统下，安装方法为：
 - ./configure --with-fortran;
 - make (或./make all, 将 cgns tool 也编译安装)
 - sudo make install (需要超级用户口令);

该命令为将头文件(*.h)拷贝到./usr/local/include 文件夹下面；将库文件（libcgns.a）拷贝到/usr/local/lib 文件夹下；

如果没有超级用户口令，也可将*.h 和 libcgns.a 拷贝到某个文件夹里面。

安装好该套件后，就可以在程序调用 cgns 库函数了，如调用 cgns 库函数，需要在连接选项中添加“-lcgns”选项。

8.3 CGNS 的边界条件代码

查阅“cgnslib_f.h”中的定义语句，CGNS 预定义的边界条件有如下 25 种：

parameter (BCAxisymmetricWedge	= 2)
parameter (BCDegenerateLine	= 3)
parameter (BCDegeneratePoint	= 4)
parameter (BCDirichlet	= 5)
parameter (BCExtrapolate	= 6)
parameter (BCFarfield	= 7)
parameter (BCGeneral	= 8)
parameter (BCInflow	= 9)
parameter (BCInflowSubsonic	= 10)
parameter (BCInflowSupersonic	= 11)
parameter (BCNeumann	= 12)
parameter (BCOutflow	= 13)
parameter (BCOutflowSubsonic	= 14)
parameter (BCOutflowSupersonic	= 15)
parameter (BCSymmetryPlane	= 16)
parameter (BCSymmetryPolar	= 17)
parameter (BCTunnelInflow	= 18)
parameter (BCTunnelOutflow	= 19)
parameter (BCWall	= 20)
parameter (BCWallInviscid	= 21)
parameter (BCWallViscous	= 22)
parameter (BCWallViscousHeatFlux	= 23)
parameter (BCWallViscousIsothermal	= 24)
parameter (FamilySpecified	= 25)

可以看出，与 Gridgen 的 Generic 格式相比，CGNS 对边界条件的描述更加细致。

第九章 后处理

§9.1 气动力、矩的计算

§9.2 其他物理量的计算

1. 速度梯度张量第二不变量 Q 的计算

速度梯度张量第二不变量

$$Q = II\left(\frac{\partial u_i}{\partial x_j}\right) = \frac{1}{2}(\Omega_{ij}\Omega_{ij} - S_{ij}S_{ij})$$

是一个常用的涡识别器。到某处 $Q>0$ 时，通常意味着该处是涡的中心。因为对于不可压流动有

$$\nabla^2 p = 2\rho Q$$

因而当某处 $Q>0$ 时，该处的压力较低，与涡心周围的压力分布相近。因而较高 Q 值 ($Q>0$) 的等值面经常被用于识别流场中的拟序涡结构。

第十章 算例

§10.1 二维程序测试算例

1 在随机网络上计算导数

测试算例：

令

$$u(x, y) = \sin(\pi x) \cos(\pi y),$$

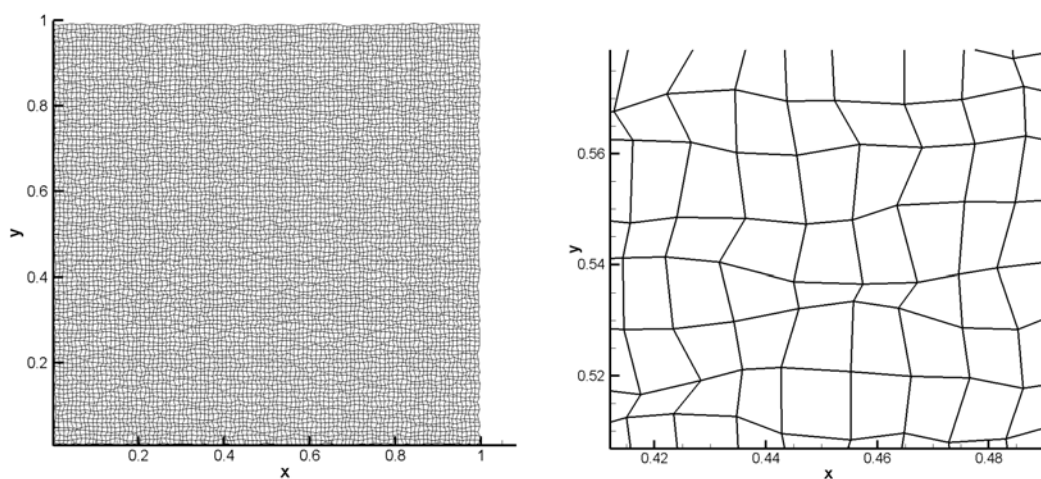
计算网格公式如下：

$$x_{i,j} = \frac{i-1}{N-1} + \frac{0.4}{N-1} r_1$$

$$y_{i,j} = \frac{j-1}{N-1} + \frac{0.4}{N-1} r_2$$

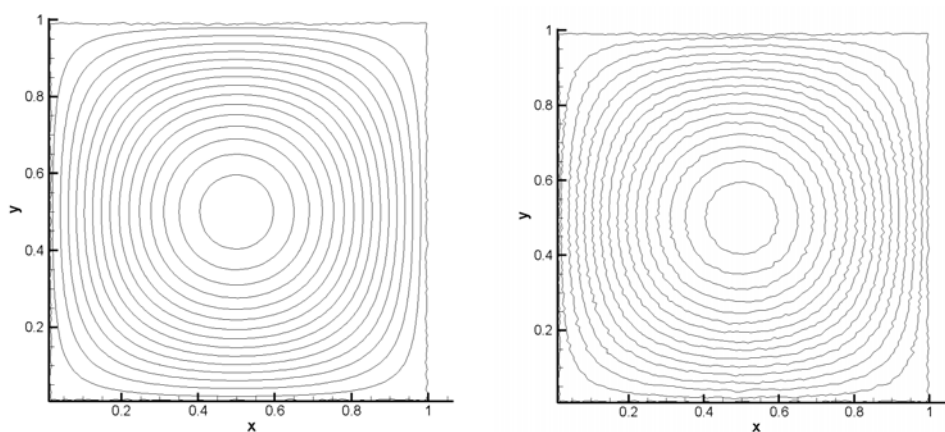
其中 r_1 和 r_2 是 $(-1,1)$ 之间的随机数，网格点 101×101 。

利用§2.2 中的方法计算偏导数 $\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}$ ，与精确解相比计算其误差。



计算网格及局部放大图

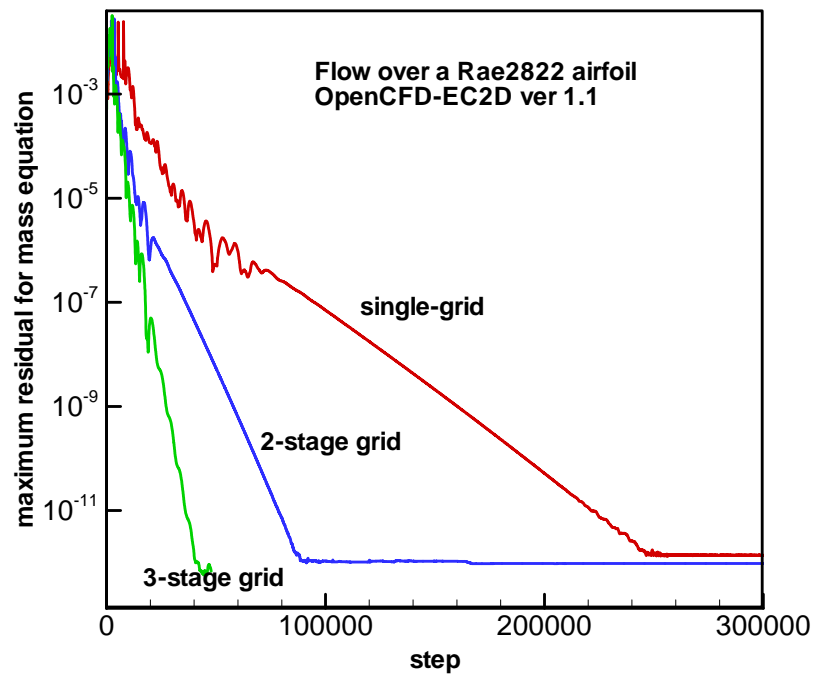
最大误差： x 方向导数： 0.077； y 方向导数 0.065



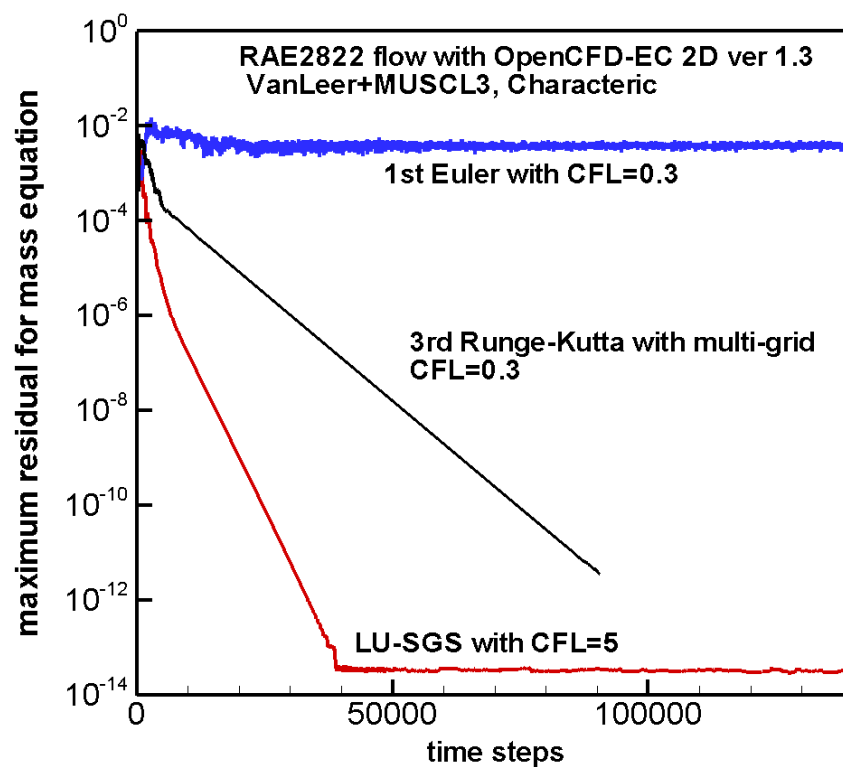
y 方向导数的精确解（左）与数值解（右）

2 RAE2822 翼型绕流

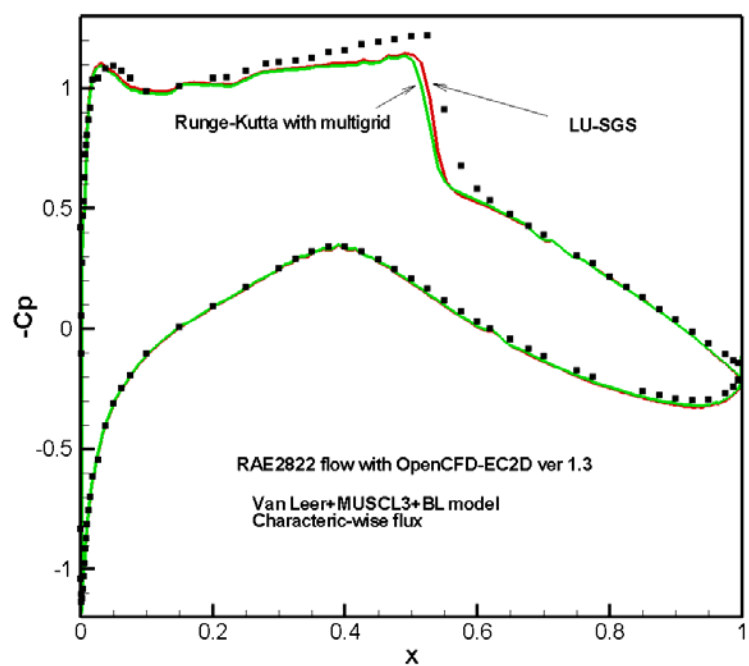
该算例是考核程序的标准算例，计算工况及网格见 NASA 网站：
<http://www.grc.nasa.gov/WWW/wind/valid/raetaf/raetaf.html>



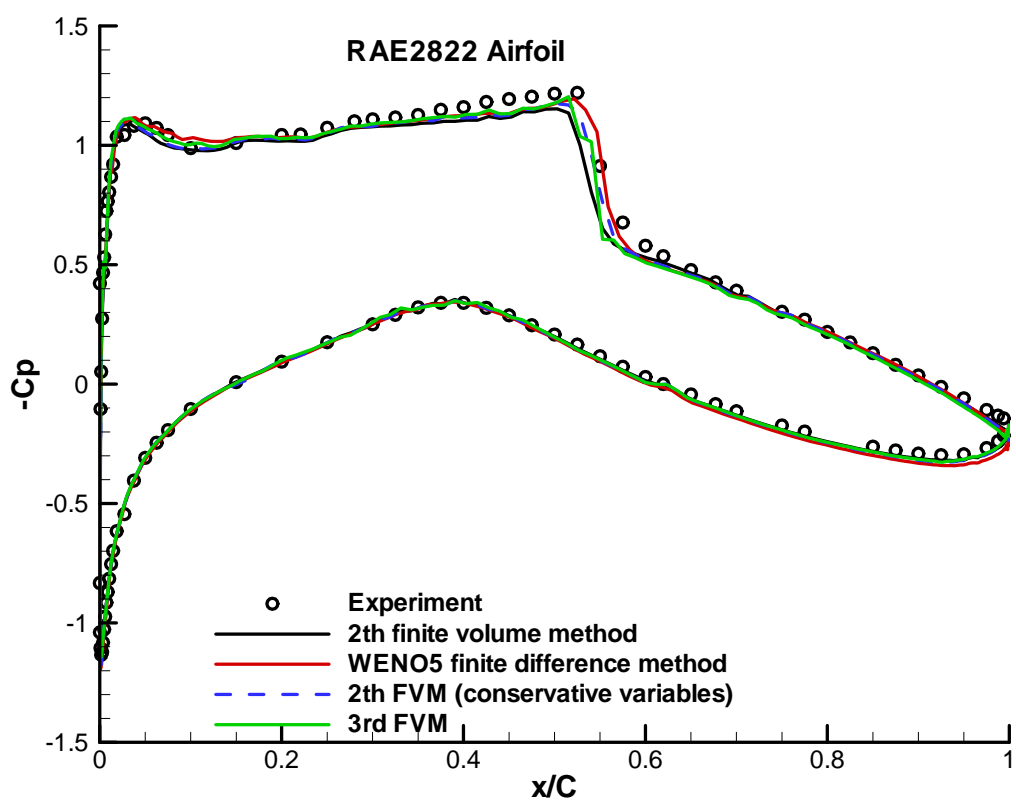
RAE2822 翼型绕流问题的残差收敛情况



RAE2822 翼型绕流问题的残差收敛情况（显格式 vs. 隐格式）

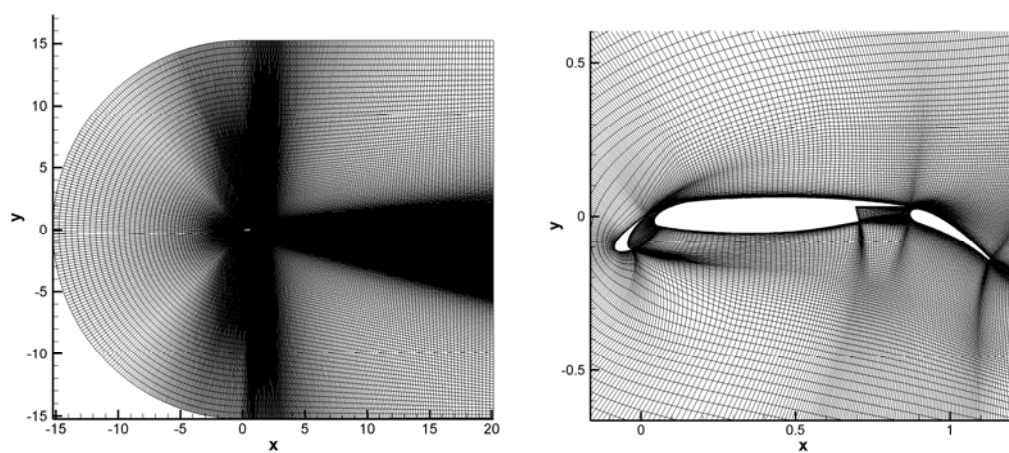


翼型表面压力分布

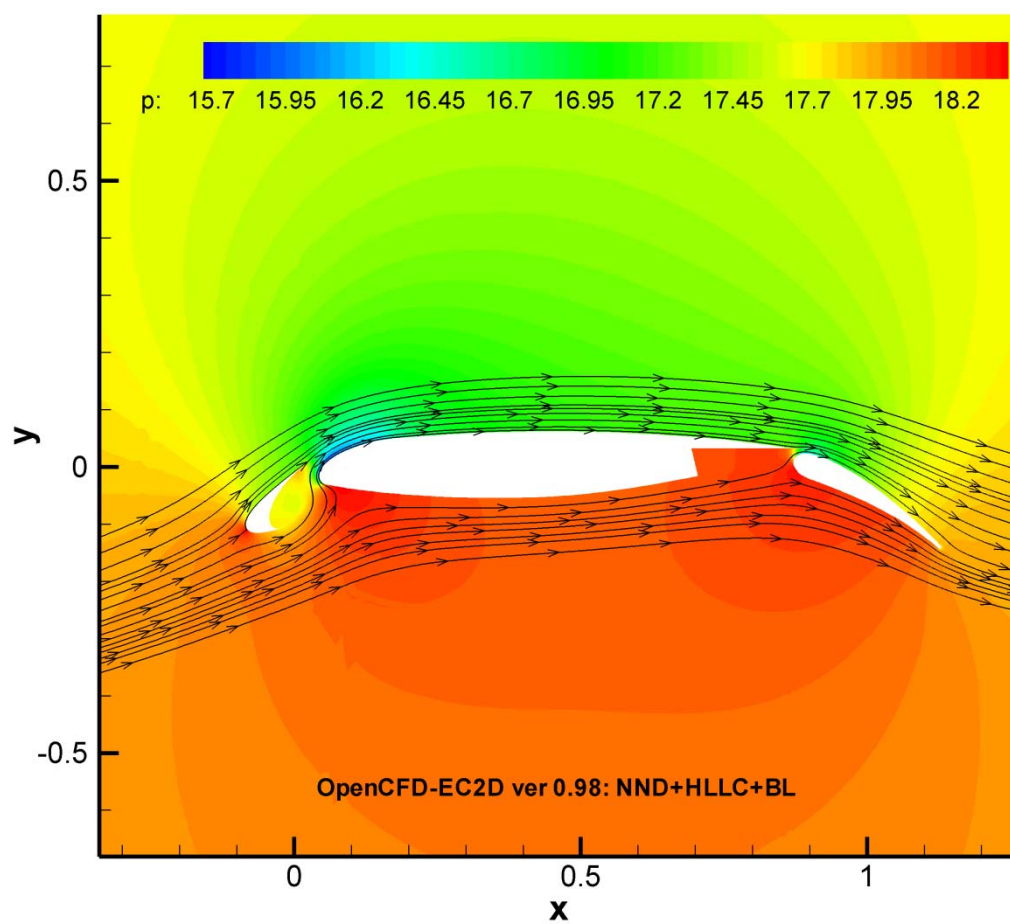


表面系数分布: OpenCFD-EC v0.98 的计算结果

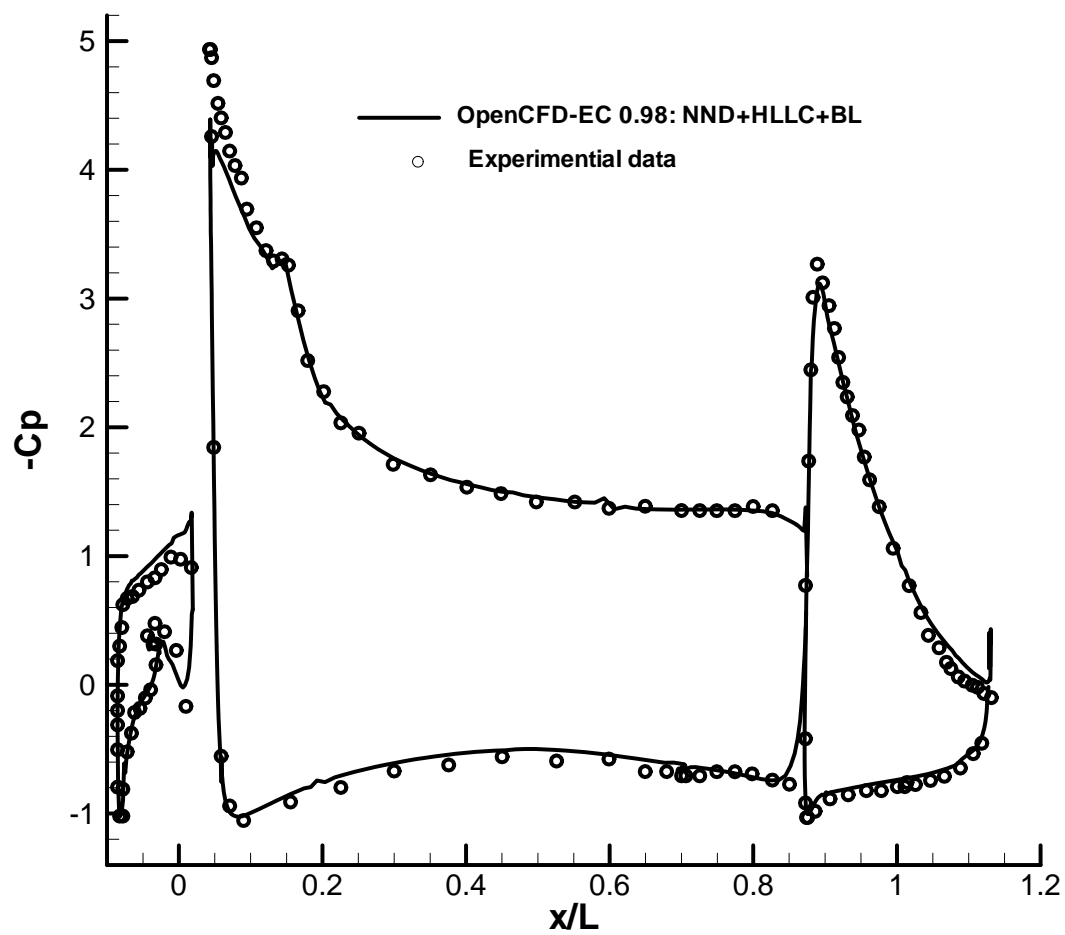
3 30P30N 三段翼型



计算网格

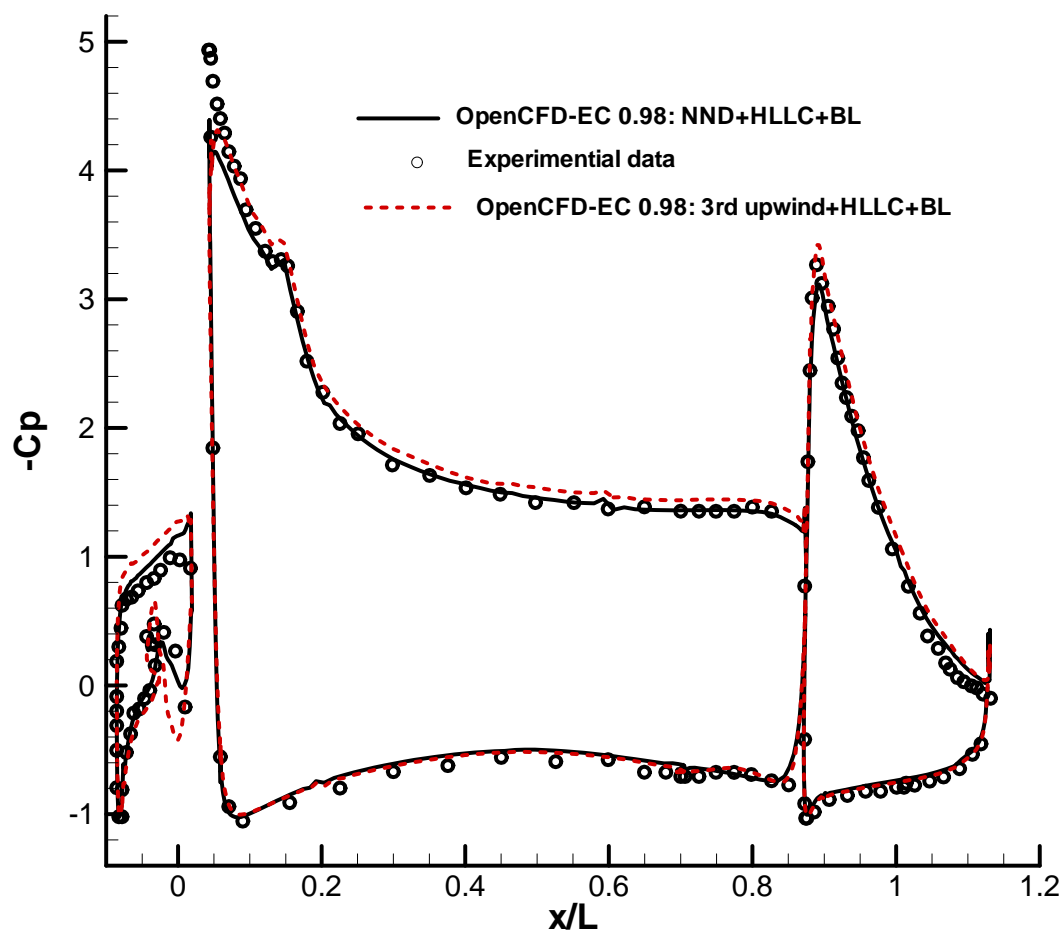


压力分布及流线



翼型表

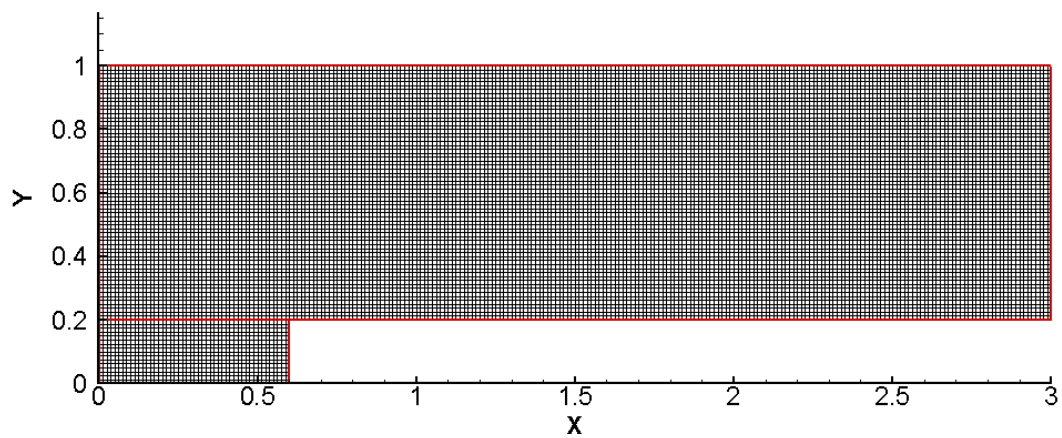
面的压力分布

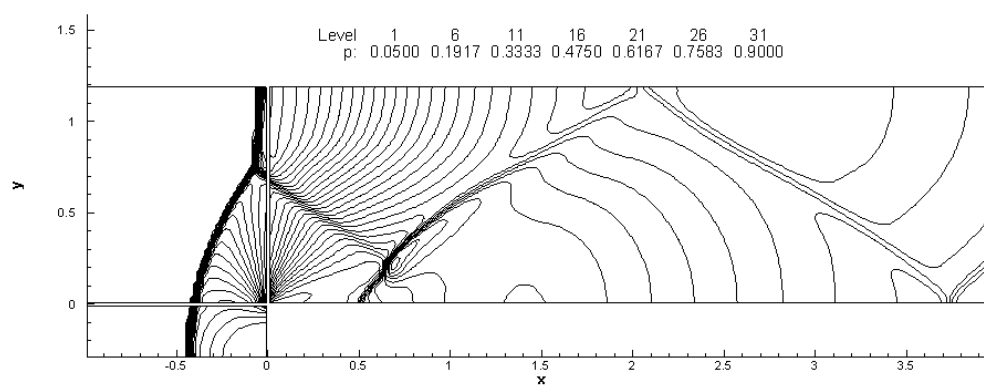


翼型表面的压力分布（三阶迎风 vs 二阶 NND）

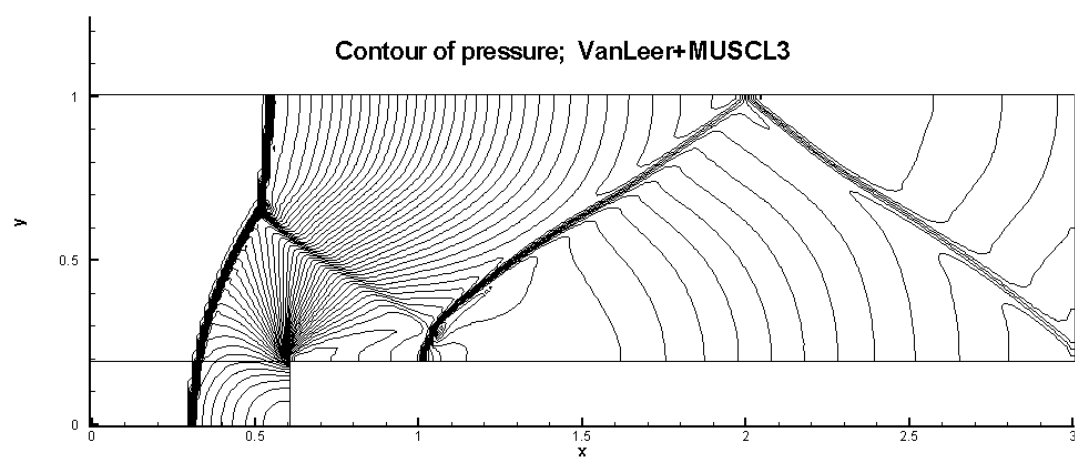
4 前台阶问题（ $Ma=3$ ）

计算网格 $\Delta = 1/80$

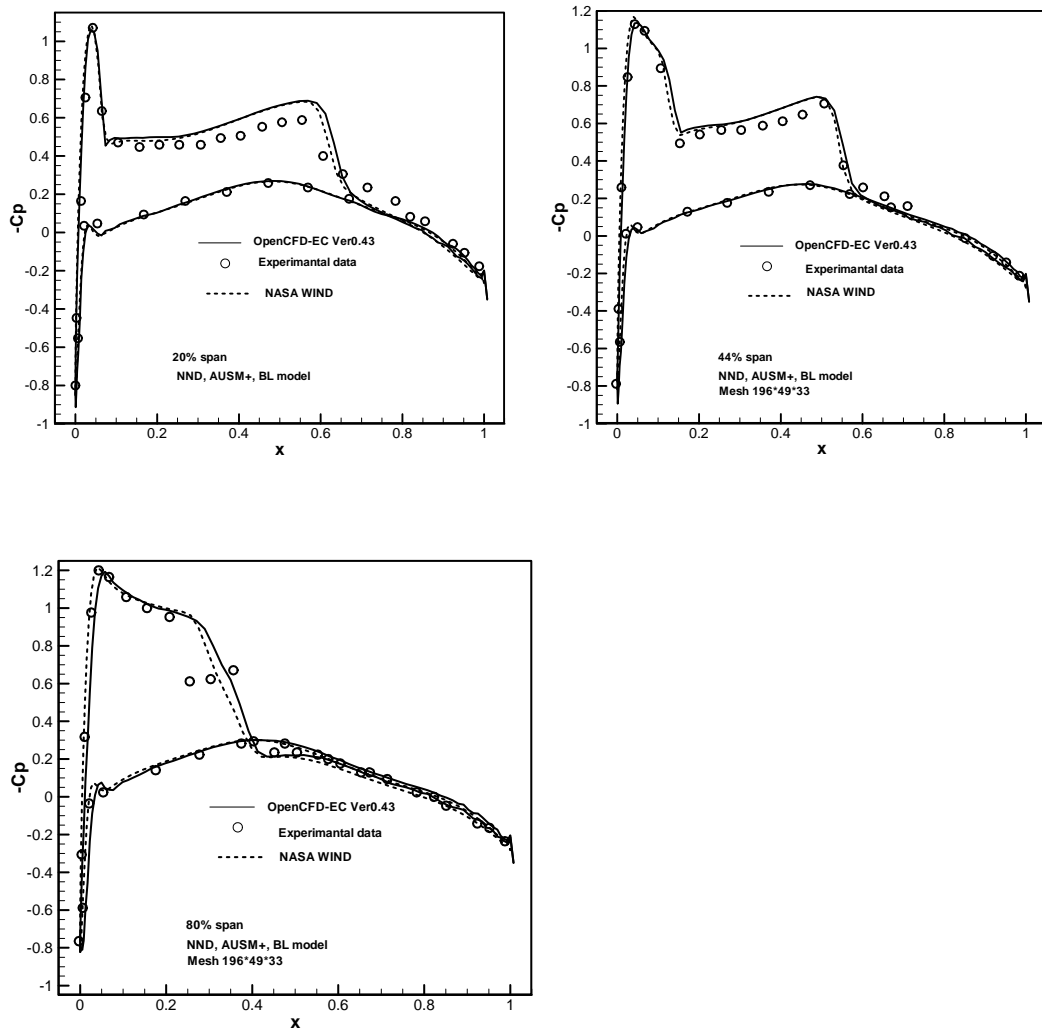




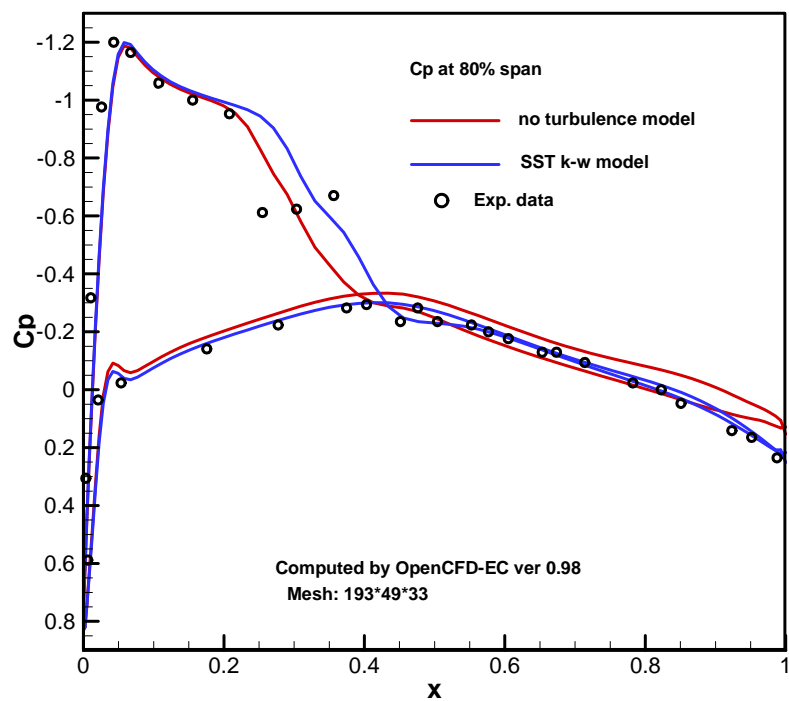
前台阶问题的压力分布



7.2 M6 翼

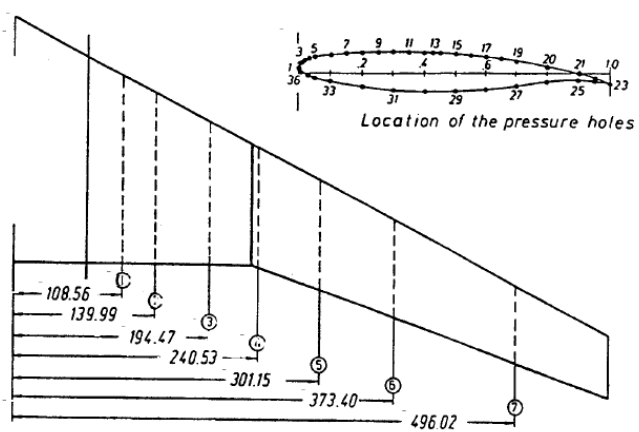


M6 翼不同位置的压强系数

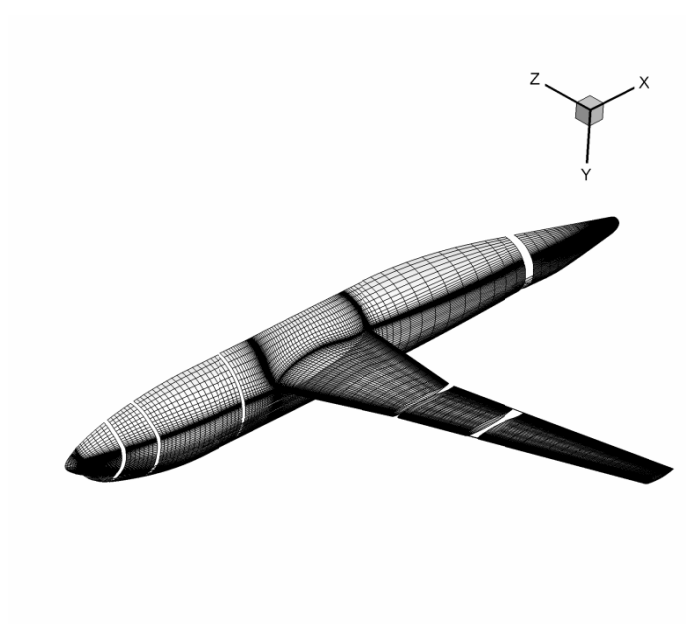


80% 翼展处的压力系数

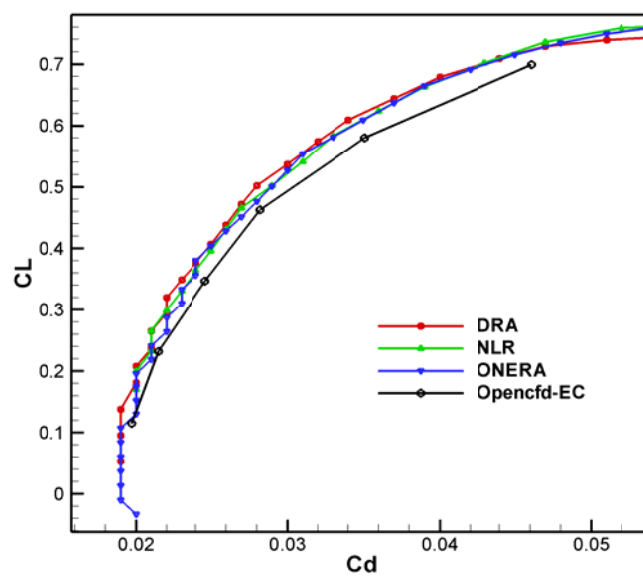
7.3 DLR-F4 翼身组合体



机翼指定横截面位置

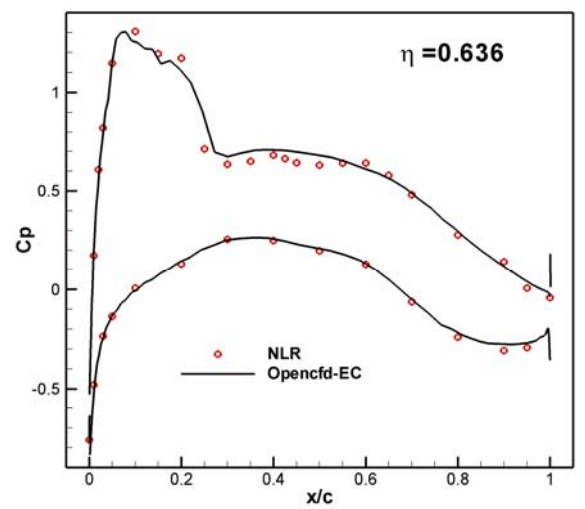
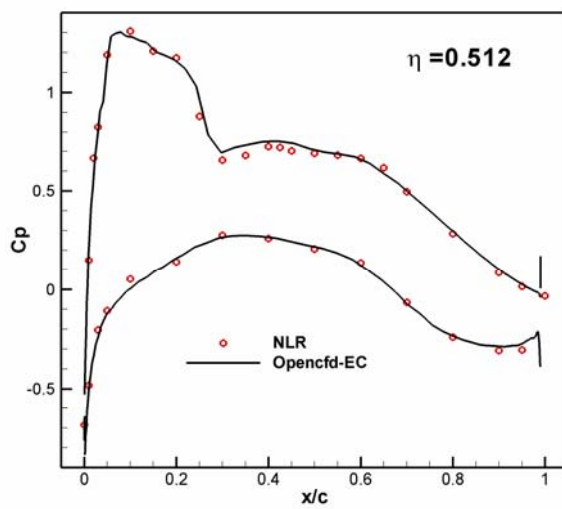
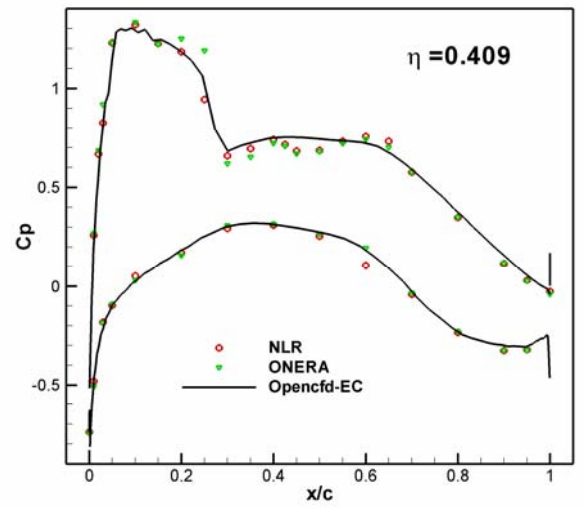
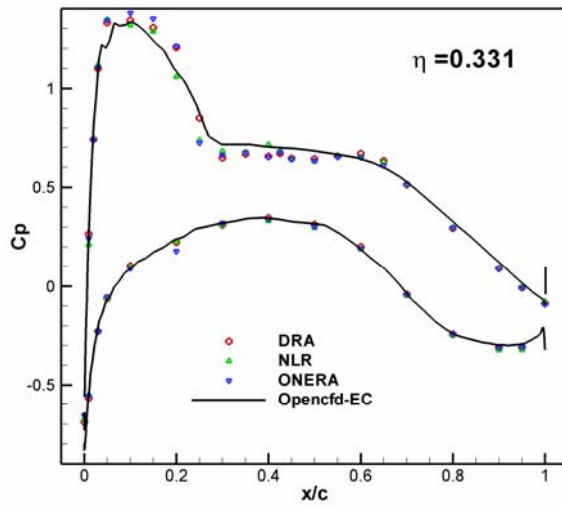
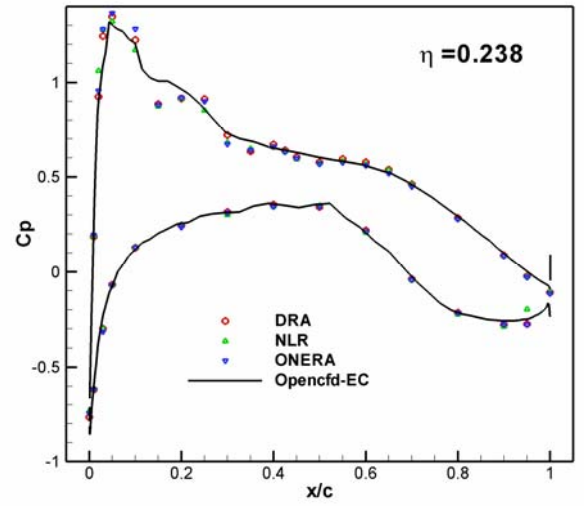
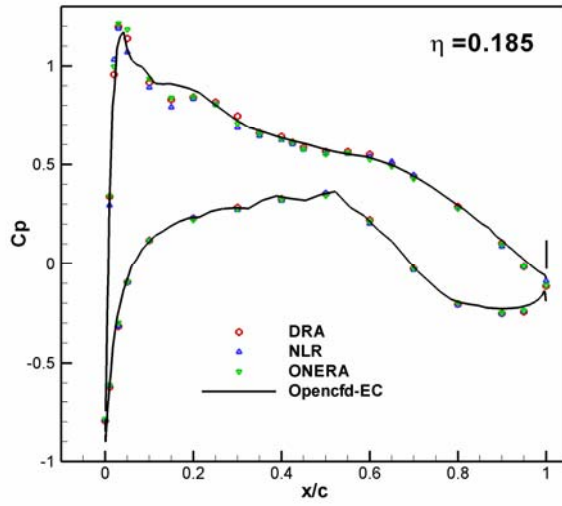


表面网格



DLR-F4翼身组合体极曲线

机翼表面不同站位的压力系数分布：



附录 1 叶轮机械内流求解器

从 1.1 版起，OpenCFD-EC 支持叶轮机械内流的数值模拟。

计算模型为叶轮机械（压气机、涡轮）叶片通道内的流动。采用直角坐标系，规定 x 方向为轴向。

在控制文件中，设定 **IF_TurboMachinery=1** 启用**叶轮机械计算模式**。该模式有如下特点：

- 1) 计算坐标 x 轴固定在叶轮机械轴上，与叶轮机械共同旋转（**旋转坐标系，非惯性系**）。
- 2) 边界条件：采用旋转叶轮机械专用的边界条件。见后文。
- 3) 流动的无量纲化有特殊约定。

1.1 控制方程 及无量纲化

坐标系建立在叶轮机的轴上（规定 **x -轴正方向与叶轮机的轴重合**），随叶轮机的轴一起转动。因而坐标系为非惯性系，流体质点受到惯性力的作用：

$$F_q = -(\vec{\omega} \times \vec{\omega} \times \vec{r} + 2\vec{\omega} \times \vec{v})$$

流动控制方程为：

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_1(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{F}_2(\mathbf{U})}{\partial y} + \frac{\partial \mathbf{F}_3(\mathbf{U})}{\partial z} = \frac{\partial \mathbf{F}_{v1}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{F}_{v2}(\mathbf{U})}{\partial y} + \frac{\partial \mathbf{F}_{v3}(\mathbf{U})}{\partial z} + S_\omega$$

其中 S_ω 为由于坐标系转动而产生的源项（惯性力）：

$$S_\omega = \begin{bmatrix} 0 \\ 0 \\ \rho(\omega^2 y + 2\omega w) \\ \rho(\omega^2 z - 2\omega v) \\ \rho\omega^2 (vy + wz) \end{bmatrix}$$

其中 ω 为叶轮机旋转的角速度。注意到在旋转坐标系中，单位体积的惯性力为：

$$\vec{f} = \rho\omega^2 \vec{r} - 2\rho\vec{\omega} \times \vec{v}。$$

其余各项的表达式与常规的 N-S 方程相同。

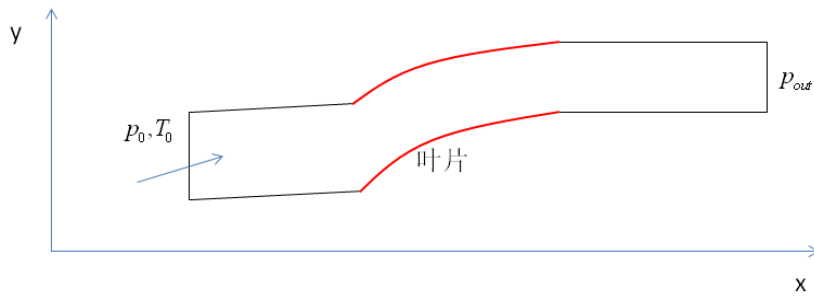


图 1 计算域示意图（平行与轴向的截面）

选取特征长度 L^R ，特征速度 u^R ，特征密度 ρ^R ，特征温度 T^R ，特征粘性系数 μ^R 将方程无量纲化。

原则上将，这种特征量的选取可以是任意的。很多 CFD 软件选择各自的单位量（例如 1m, 1m/s, 1Kg/m³）作为特征量(实际上是有量纲计算)。

本求解器面向叶轮机内的流场计算。通常情况下，计算条件是给定入口处的总温 T_0 ，总压 P_0 以及出口处给定静压 P_{out} ，通过实际计算得到出口处的流速及其他流动条件。

由于计算之前并不知道入口处的实际流速，因此不宜以入口处的物理量作为特征量进行无量纲化。

因此，选择滞止条件下的物理量作为特征量：

特征温度： $T^R = T_0$

特征密度： $\rho^R = \rho_0 = P_0 / (RT_0)$

特征速度：选取入口滞止条件下的声速 $u^R = a_0 = \sqrt{\gamma P_0 / \rho_0}$ 。因此，特征 Mach 数（计算 Mach 数）为 1。

特征粘性系数：选取特征温度的气体粘性系数，这是一个物性参数。

特征压力： $p^R = \rho_0 a_0^2$

特征长度：本软件不做特别要求。特征长度 L^R 为网格长度 1 对应的实际长度。例如网格文件中单位 1 对应 1mm，则特征长度 L^R 为 1mm。如果网格文件中单位 1 为某特殊尺度（例如叶片的弦长），则 L^R 也为该尺度。

其中，叶轮及旋转角速度 ω 是以特征速度及特征长度无量纲化：

$$\omega = \frac{\tilde{\omega}}{u^R / L^R} = \frac{\tilde{\omega}}{a_0 / L^R}$$

其中， $\tilde{\omega}$ 为有量纲角速度（控制文件中，单位为 转/秒）。选用该方式无量纲化，源项 S_ω 中不会出现新的无量纲数。

1.2 边界条件

a. 进口边界条件：对于轴向亚音速来流，需给定四个流动参数。本文给定入口处的**总压和总温**。另外假设该压气机转子**轴向进气**，即入口处的绝对速度方向为轴向，这样入口处的两个绝对进气角就给定了。

例如，通过内点外插获得了入口处的静压 p_{in} ，由等熵关系可获得入口处的温度、密度及轴向速度：

$$\begin{aligned} T_{in} / T_0 &= (p_{in} / p_0)^{\frac{\gamma-1}{\gamma}} \\ \rho_{in} &= p_{in} / (RT_{in}) \\ v &= \sqrt{2c_p(T_0 - T_{in})} \end{aligned}$$

在本无量纲方式下，

$$R = \frac{1}{\gamma Ma^2} = \frac{1}{\gamma}, \quad c_p = \frac{1}{(\gamma-1)Ma^2} = \frac{1}{\gamma-1}$$

b. 出口边界条件：对于轴向亚音速出流，需给出一个流动参数，本文给定为出口处的静压。叶轮机中的流动为旋转状态，由于离心力的影响，存在径向的压力梯度，因而，指定出口处的压力控制方程为：

$$\frac{\partial p}{\partial r} = \bar{\rho} \bar{v}_\theta^2 / r$$

其中 $\bar{\rho}, \bar{v}_\theta$ 分别为出口处沿周向平均后的密度及周向速度（**绝对速度**，即相对速度叠加上由于坐标旋转带来的牵连速度）。

指定轮毂处的背压 p_{out}^h ，则出口平面内的压力可由上式积分得到。

c. 壁面边界条件：在叶片表面、轮毂面和机匣面等固体壁面上，我们采用无滑移条件。对于温度采用绝热条件 $\partial T / \partial n = 0$ 。

d. 周期性边界条件：在周期性区域的对应边界点处，流动应当满足周期性条件。应当注意到速度场的周期性条件是指速度的轴向分量、周向分量及法向分量满足周期性条件，而速度矢量本身并不满足周期性条件（如图 2 所示）。

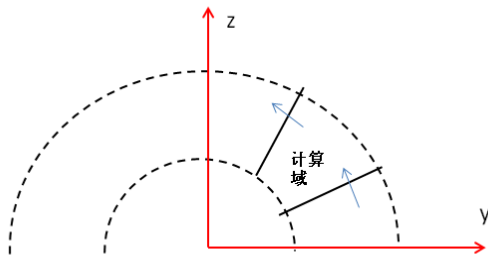


图 2 周期性边界条件示意图（垂直于轴向的截面）

参考文献:

T. Arima, T. Sonoda, M. Shirotori et al. A numerical investigation of transonic axial compressor rotor flow using a low-Reynolds-number $k-\epsilon$ turbulent model, Journal of Turbomachinery, 121,44-58,1999