

椭圆型方程曲线网络的快速生成方法*

吴晋湘 王振国 庄逢辰

(航天技术系)

摘要 本文在用椭圆型方程的网格生成方法生成曲线网格时,通过比较四种常用的计算格式,得到了收敛速度较快的生成方法。在计算流体力学领域,对于边界形状随时间变化的流动问题的数值计算,及自适应网格的生成等问题,生成网格的收敛速度直接影响流场的计算速度。在上述情况下使用本文的结果,能有效的减少机时,提高计算效率。

关键词 网格生成,数值计算

分类号 O242.21

曲线网格生成技术的发展,对有限差分方法在计算流体动力学领域中的应用有着十分重要的意义。目前已发展有多种数值曲线网络的生成方法^{[1][3]}其中以保角变换法和椭圆型函数法较为常用。对于一些比较特殊的边界形状,保角变换法可以直接生成正交的曲线网格。文献[6]进一步给出了一般形状的计算域的保角变换网格生成法。但目前仍以 J. F. Thompson 等提出的椭圆型方程网格生成方法更为常用,它不仅能处理二维和三维的问题,定常和非定常的问题,还能正确的表达边界形状,调整网格密度,生成的网格质量也很高。

对于大多数的计算流体动力学问题,一般在曲线网格生成后,再在计算网格域内数值求解其动力学控制方程组。相对后一部分工作,网格生成的计算工作量要小得多。但对于一些特殊的情况,如计算域边界随时间而变化,或如因在激波附近、剪切层附近加密网格时,需要所生成的网格随流场的计算过程变化而变化,这时网格生成的收敛速度对整个计算机时的影响就会变得较大,因而希望有一种收敛速度较快的网格生成方法。

1 数学模型

Thompson 法即是利用如 Laplace 方程或 Poisson 方程这样的椭圆型方程,来建立坐标变换关系式,在物理平面上生成贴体的曲线网格。利用 Laplace 方程作为变换关系,可将计算平面 (ξ, η) 内的平行直线正交网格变换到物理平面 (x, y) 内,其变换关系式如下:

$$\xi_{xx} + \xi_{yy} = 0 \quad (1)$$

$$\eta_{xx} + \eta_{yy} = 0 \quad (2)$$

* 1992年4月10日收稿

这种变换既能正确地表达边界形状,也能使物理平面内的曲线、网格的正交性基本得到满足。如在(1),(2)式中分别加上源项 $P(\xi, \eta), Q(\xi, \eta)$, 则构成了 Poisson 方程变换式:

$$\xi_{xx} + \xi_{yy} = P(\xi, \eta) \quad (3)$$

$$\eta_{xx} + \eta_{yy} = Q(\xi, \eta) \quad (4)$$

这样虽然使物理平面内曲线网格的正交性有所破坏,但由于通过调整函数 P 和 Q , 可以对网格的密度进行调整和控制, 因此对计算存在局部大梯度物理量的流场具有十分明显的优势。

在实际生成网格的过程中, 由于物理平面的网格点是未知的, 直接求解(1)、(2)或(3)、(4)式并不方便。可以利用坐标量的变换关系式:

$$\begin{aligned} \xi_x &= y_\eta/J, & \xi_y &= -x_\eta/J \\ \eta_x &= -y_\xi/J, & \eta_y &= x_\xi/J \\ J &= x_\xi y_\eta - x_\eta y_\xi \end{aligned} \quad (5)$$

很容易的求出以上变换关系的逆变换, 如 Poisson 方程的逆变换为:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2(x_\xi P + x_\eta Q) \quad (6)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2(y_\xi P + y_\eta Q) \quad (7)$$

式中, $\alpha = x_\eta^2 + y_\eta^2; \beta = x_\xi x_\eta + y_\xi y_\eta; \gamma = x_\xi^2 + y_\xi^2$

(6),(7)式中如令 $P=Q=0$, 则得 Laplace 变换的逆变换:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0 \quad (8)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0 \quad (9)$$

对变换关系式(6),(7)或(8),(9)进行差分求解已没有原则困难, 可以生成令人满意的曲线网格。但对于如计算边界是随时间而变化的非定常问题, 或如自适应网格的生成, 在超音速流的计算中, 由于在计算前并不能准确地知道激波的位置, 网格的加密区域要随着激波的捕获过程而逐步明朗, 这时网格的形式要随着流场的计算而不断变化。在这些情况下, 如每一次生成网格所占的时间较长, 将大大增加整个流场的计算时间。如前所述, 这时将希望有一种收敛速度比较快的差分格式。

2 网格生成的差分方法讨论

本文以 Laplace 逆变换方程(8)、(9)为例, 讨论了几种常用差分方法用于计算网格生成的收敛速度。由于(8)式和(9)式在形式上是对称的, 本文以下将以(8)式为例进行分析。令计算平面内的网格步长 $\Delta\xi = \Delta\eta = 1$, (8)式的差分形式可以写成:

$$A_r x_{i+1,j} + A_w x_{i-1,j} + A_c x_{i,j} + A_n x_{i,j+1} + A_s x_{i,j-1} = S_{i,j} \quad (10)$$

式中, $A_r = A_w = \frac{1}{4} [(x_{i,j+1} - x_{i,j-1})^2 + (y_{i,j+1} - y_{i,j-1})^2]; A_n = A_s = \frac{1}{4} [(x_{i+1,j} - x_{i-1,j})^2 + (y_{i+1,j} - y_{i-1,j})^2];$

$$A_i = -(A_r + A_w + A_n + A_s)$$

$$\begin{aligned} S_{i,j} &= \frac{1}{8} [(x_{i+1,j} - x_{i-1,j})(x_{i,j+1} - x_{i,j-1}) + (y_{i+1,j} - y_{i-1,j})(y_{i,j+1} - y_{i,j-1})] \\ &\quad \cdot (x_{i+1,j+1} - x_{i-1,j+1} - x_{i+1,j-1} + x_{i-1,j-1}) \end{aligned}$$

2.1 简单迭代法 (SIM)

简单迭代法是在差分求解(10)式的过程中,用周围四点上一次的值求计算点的新值,通过迭代直至收敛。即:

$$x_{i,j}^{n+1} = \frac{1}{A_c}(S_{i,j} - A_e x_{i+1,j}^n - A_w x_{i-1,j}^n - A_n x_{i,j+1}^n - A_s x_{i,j-1}^n) \quad (11)$$

2.2 高斯-赛德尔迭代法加超松弛 (GSS)

在简单迭代法中,如果已扫描过的两点用新值取代旧值,即成高斯-赛德尔迭代法。这里(11)式可以写成:

$$x_{i,j}^{n+1} = \frac{1}{A_c}(S_{i,j} - A_e x_{i+1,j}^n - A_w x_{i-1,j}^{n+1} - A_n x_{i,j+1}^n - A_s x_{i,j-1}^{n+1}) \quad (12)$$

再在计算结果 $x_{i,j}^{n+1}$ 和 $x_{i,j}^n$ 之间乘一个大于 1 小于 2 的加权因子进行平均,就是超松弛 (SOR):

$$x_{i,j}^{n+1} = \omega x_{i,j}^{n+1} + (1 - \omega)x_{i,j}^n \quad (1 < \omega < 2) \quad (13)$$

ω 为松弛因子。最佳超松弛因子按下式求取^[4]:

$$(\omega_{opt})_{i,j} = \frac{2}{1 + \sqrt{1 - \rho_{i,j}^2}} \quad (14)$$

$$\rho_{i,j} = \frac{1}{\alpha_{i,j} + \gamma_{i,j}} \left[\alpha_{i,j} \cos\left(\frac{\pi}{i_{max} - 1}\right) + \gamma_{i,j} \cos\left(\frac{\pi}{j_{max} - 1}\right) \right]$$

式中 $\alpha_{i,j}$ 和 $\gamma_{i,j}$ 即是 (ij) 点的 A_e 和 A_n , 以这种方法给出的最佳超松弛因子在计算域内各点不同。

2.3 交替方向隐式迭代法 (ADI)

交替方向隐式迭代法的思想是先在 ξ 方向隐式求解左、中、右三点的新值,再在 η 方向隐式求解上、中、下三点的新值。在 ξ 方向求解时,已扫描过的下方的 $x_{i,j-1}$ 也可用新值取代。其迭代过程为:

$$A_e x_{i+1/2,j}^{n+1/2} + A_c x_{i,j}^{n+1/2} + A_w x_{i-1/2,j}^{n+1/2} = S_{i,j} - A_n x_{i,j+1}^n - A_s x_{i,j-1/2}^{n+1/2} \quad (15)$$

$$A_n x_{i,j+1}^{n+1/2} + A_c x_{i,j}^{n+1/2} + A_s x_{i,j-1}^{n+1/2} = S_{i,j} - A_e x_{i+1/2,j}^{n+1/2} - A_w x_{i-1/2,j}^{n+1/2} \quad (16)$$

这样,在隐式求解方向需用追赶法 (TDMA) 求解一个三对解矩阵。对计算结果进行超松弛处理,最佳松弛因子可按式给出^[5]:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho^2}} \quad (17)$$

$$\rho = \frac{1}{2} \left[\cos\left(\frac{\pi}{i_{max} - 1}\right) + \cos\left(\frac{\pi}{j_{max} - 1}\right) \right]$$

2.4 强隐式迭代法 (SIP)

如果计算点和周围四点都用新值,即:

$$A_e x_{i+1,j}^{n+1} + A_w x_{i-1,j}^{n+1} + A_c x_{i,j}^{n+1} + A_n x_{i,j+1}^{n+1} + A_s x_{i,j-1}^{n+1} = S_{i,j} \quad (18)$$

其矩阵形式可以写成:

$$[A]\{X\} = \{S\} \quad (19)$$

系数矩阵 $[A]$ 是一个五对角矩阵。直接求解五对角矩阵 $[A]$ 非常困难,强隐迭代法首

先将系统矩阵分解为一个下三角阵 $[L]$ 和一个上三角阵 $[U]$ ，这两个矩阵的乘积是一个近似于 $[A]$ 的七对角矩阵。即：

$$[L][U] = [A'] \approx [A]$$

将关系式(19)写成：

$$[A']\{X^{n+1}\} = [A']\{X^n\} + \{S\} - [A]\{X^n\}$$

即有：

$$[A']\{\delta^{n+1}\} = \{S\} - [A]\{X^n\} \quad (20)$$

求出 $\{\delta^{n+1}\}$ 后，再利用关系：

$$\{X^{n+1}\} = \{\delta^{n+1}\} + \{X^n\} \quad (21)$$

求出 x_i^{n+1} 。SIP 法的收敛速度由因子 α 控制，详细方法可参阅文献[7]。

3 算例

本文所选用的第一个算例的应用背景是数值计算单个液滴在气相流场中的变形和破碎过程。球形液滴在有相对流速的气流中，由于气动力、表面张力和粘性力的相互作用，会逐渐变形甚至破碎。在用数值计算的方法模拟这一过程时，如选用贴体曲线坐标，网格系统应在计算过程中随液滴表面形状的变化而变化。初始气相场的计算边界即是图 4 所生成的网格的边界。对于同样的边界情况，本文对 31×11 网格点和 61×21 网格点的曲线网格生成进行了计算，其中 31×11 网格点的生成网格如图 4 所示。

另一个算例的边界情况比较复杂，其应用背景是同轴式喷嘴气相流场的数值计算。选用此算例的主要目的是考察各种算法的计算稳定性。生成的网格如图 5 所示，图 6 是喷嘴出口的计算域头部网格局部放大。

图 1 至图 5 分别表示了三种情况下，四种不同算法的收敛过程。其横坐标为迭代次数 N ，纵坐标为最大相对误差 ϵ ：

$$\epsilon = \max \left(\left| \frac{x_{i,j}^{n+1} - x_{i,j}^n}{x_{i,j}^n} \right|, \left| \frac{y_{i,j}^{n+1} - y_{i,j}^n}{y_{i,j}^n} \right| \right) \quad (22)$$

$$i = 2, 3, \dots, i_{\max} - 1; j = 2, 3, \dots, j_{\max} - 1$$

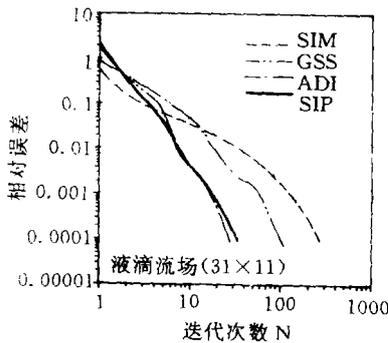


图 1 四种数值方法收敛速度的比较 (A)

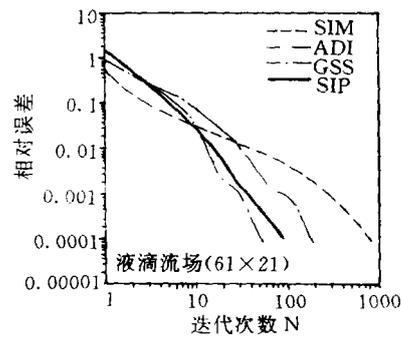


图 2 四种数值方法收敛速度的比较 (B)

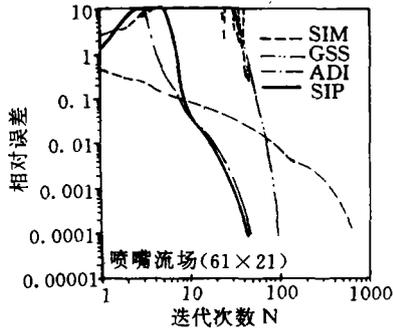


图3 四种数值方法收敛速度的比较 (C)

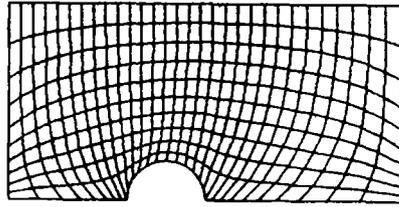


图4 液滴周围流场计算域的网格生成 (SIP法)

收敛判据为： $\epsilon \leq 10^{-4}$ 。四种算法最后都收敛到同相的结果，以SIM法为基准的最大相对误差小于0.02。

由图1至图3可知，达到同样精度的收敛条件，ADI法、SIP法和GSS法的迭代次数时显少于SIM法。在这三种算法中，又以ADI和SIP两种算法的迭代次数更少。但由于后两种算法每一步的计算工作量较大。实际节省机时的效果不如迭代次数的减少那样明显（见表1），但仍不失为一种可取的算法。尤其对于算例一的情况，ADI法的优势更为明显。GSS法编制程序简单，收敛速度较SIM法也有明显提高，但对于边界较复杂的第二种算例稳定性较差。这表明当物理平面内的计算域边界较复杂时，应适当减小超松弛因子（对于SIP法应减小因子 α ）使 $1 < \omega < \omega_{opt}$ 。

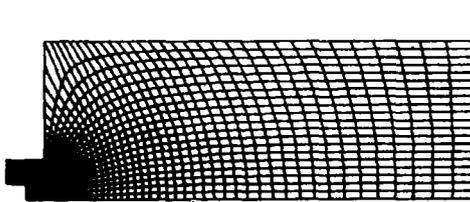


图5 喷嘴出口流场计算域的网格生成 (GSS法)

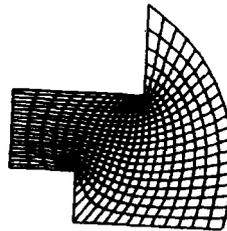


图6 图5头部区域的放大

表1 利用四种差分格式对算例进行试算的有关结果

	液滴流场(31×11)		液滴流场(61×21)		喷嘴流场(61×21)	
	迭代次数	机时(286)	迭代次数	机时(286)	迭代次数	机时(386)
SIM	236($\omega=0$)	5'19.8"	768($\omega=0$)	62'22.3"	658($\omega=0$)	1'37.6"
GSS	102($\omega=opt$)	2'39.1"	183($\omega=opt$)	18'59.1"	96($\omega=opt$)	22.2"
ADI	29($\omega=opt$)	1'25.2"	52($\omega=opt$)	10'12.4"	49($\omega=1.4$)	20.8"
SIP	34($\alpha=0.95$)	1'30.9"	90($\alpha=0.9$)	15'58.4"	45($\alpha=0.86$)	19.7"

由图4至图6可知，用Laplace方程变换式生成的网格，在计算域内部有较好的正交性。

但在边界上,由于边界点一般是按第一类边界条件强制给出的,会破坏其正交性。可利用 Poisson 变换方程加密边界附近的网格。文献[8]也给出了在边界附近提高网格正交性的方法。

由本文可知,Thompson 法能生成令人满意的计算网格。ADI 格式在提高网格生成的收敛速度方面有明显的优势。

参 考 文 献

- 1 Thompson J F. Numerical Grid Generation, Proceedings of a Symposium on the Numerical Generation of Curvilinear Coordinate Systems and Their Use in the Numerical Solution of P. D. E., New York; North-Holland. 1982
- 2 马铁犹. 计算流体力学. 北京: 京航空航天大学出版社. 1986
- 3 忻孝康等. 计算流体力学. 长沙: 国防科大出版社. 1989
- 4 方丁酉. 两相流体力学. 长沙. 国防科大出版社. 1988
- 5 郭宽良. 数值计算传热学. 安徽. 安徽科技出版社. 1987
- 6 Ives D C, Zacharias R M. Conformal Mapping and Orthogonal Grid Generation J. Propulsion, 1989; 5 (3)
- 7 Stone H L. Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations. SIAM J. Numer. Anal. 1968, 5 (3)
- 8 Luh C C. Algebraic Grid Generation with Orthogonality Control. AIAA-90-0330, 1990

The Fast Grid Generation Techniques of Elliptic Equation

Wu Jinxiang Wang Zhenguo Zhuang Fengchen
(Department of Aerospace Technology)

Abstract

In this paper, the grid generation techniques of elliptic equation are employed to generate curvilinear grids and the best difference scheme is obtained which allows fast convergence in numerical calculation by comparing four commonly used schemes. The conclusion is useful to the numerical calculation of flow, and the boundary shape is dependent on time and the adaptive grid generation.

Key words grid generation, numerical calculation