

三维欧拉流动模拟的基于伴随方程的网格自适应^{*}

◎杨 凯 周春华/南京航空航天大学

航空工程设计中,通常需要数值模拟给出较高精度的气动力计算结果,而目标函数的误差往往是未知的。在一套极密的网格上求解控制方程当然可以保证计算精度,然而这会造成极大的计算量。自适应网格技术通过局部加密,细化关键区域的网格,保证计算精度的同时提高了计算效率。

网格自适应处理的准则通常是基于流场梯度构建的,即在流场变化较快的地方进行局部加密。这将导致包含诸如激波、驻点、旋涡等物理现象的区域的网格被加密。事实上,持续不断地加密此类区域并不能总是非常有效地提高目标函数(如升力、阻力等)的计算准确性。

基于伴随方程的网格自适应是一种直接针对目标函数的自适应方法。该方法通过伴随方程将流场误差与目标函数误差联系起来,将目标函数修正项的剩余误差作为误差指示器。与基于流场特征的自适应相比,该方法避免了在流场梯度较大的部分区域进行过细的加密,而更多地加密对目标函数有直接影响的区域。

Pierce 等人首次在有限体积法中引入基于伴随方法的误差修正^[1]。Venditti 等人^[2-5]分别运用该伴随方法进行了二维无黏、有黏流动的数值模拟。Park 将 Venditti 等人^[6]伴随方法扩展到三维欧拉流动。

本文基于伴随理论和三维网格自适应技术,实现了基于三维 Euler 方程的网格自适应和误差修正方法。通过对 ONERA-M6 机翼绕流开展的数值实验,验证了本文方法的有效性。

1 数值方法

1.1 Euler 流的数值解法

控制方程为三维定常 Euler 方程,采用有限体积 median-dual 格点格式进行控制方程的空间近似。对流项的离散应用中心格式,并引入人工耗散,获得的半离散方程为

$$V_i \frac{dw_i}{dt} + Q_i - D_i = 0 \quad i = 1, N \quad (1)$$

式中: V_i 表示 i 点相应控制体的体积, w_i 为守恒变量矢量, Q_i 和 D_i 分别为对流项和人工耗散项, N 为节点总数。

标量人工耗散^[7]构造为

$$D_i = \sum_{j \in N_i} [\alpha_{ij} \epsilon_{ij}^{(2)} (w_j - w_i) \Delta S_{ij} - \alpha_{ij} \epsilon_{ij}^{(4)} (\nabla^2 w_j - \nabla^2 w_i) \Delta S_{ij}] \quad (2)$$

式中: N_i 代表节点 i 的相邻点, α_{ij} 为谱半径, $\epsilon_{ij}^{(2)}$ 和 $\epsilon_{ij}^{(4)}$ 分别为二、四阶耗散系数, ΔS_{ij} 为控制体 i 和控制体 j 公共面的面积。本文采用耗散矩阵^[8] $|\alpha_{ij}|$ 代替谱半径 α_{ij} , 用拟拉普拉斯算子^[9] $L(w_i)$ 代替双调和算子 $\nabla^2 w_i$ 。人工耗散项变为

$$D_i = \sum_{j \in N_i} [|\alpha_{ij}| \epsilon_{ij}^{(2)} (w_j - w_i) \Delta S_{ij} - |\alpha_{ij}| \epsilon_{ij}^{(4)} (L(w_j) - L(w_i)) \Delta S_{ij}] \quad (3)$$

与标准耗散相比,矩阵耗散的主要思想是引入对流通量的雅克比矩阵,代替标量 α_{ij} 。这样数值耗散与特征值直接相关,使耗散项更接近迎风格式,减小了数值耗散。拟拉普拉斯算子在任意网格上都为二阶精度,提供了很强的收敛性,允许网格存在畸形的四面体单元。

1.2 伴随方程的求解及目标函数的误差估算和修正

自适应过程中, Euler 方程求解结束后,需求解伴随方程得到伴随解。伴随解将当地流动残值与目标函数误差联系起来,可用于误差估算和构造自适应参数。记伴随解为 Ψ , 伴随方程为^[10]

$$\left[\frac{\partial R}{\partial w} \right]^T \Psi = \left[\frac{\partial f}{\partial w} \right]^T \quad (4)$$

式中: $R=Q-D$ 表示流动方程的残值, f 表示目标函数。伴随方程难以直接求解,在方程中加入虚拟时间导数项,采用时间推进的方法获得伴随解。加入时间项的伴随方程离散形式为

$$V_i \frac{d\Psi_i}{dt} + \left(\left[\frac{\partial R}{\partial w} \right]^T \Psi \right)_i = \left(\left[\frac{\partial f}{\partial w} \right]^T \right)_i \quad i = 1, N \quad (5)$$

* 基金项目:国家自然科学基金(11472134)资助项目。

用 \mathbf{R}_i^p 表示伴随方程的残值,式(5)变为

$$V_i \frac{d\boldsymbol{\Psi}_i}{dt} + \mathbf{R}_i^p = 0 \quad i = 1, N \quad (6)$$

伴随方程与流动方程形式一致,部分表达式和推导可见文献[10],唯一不同的是人工耗散的相关项。本文人工耗散对应的伴随求解表达式为

$$\left(\left(\frac{\partial \mathbf{D}}{\partial \mathbf{w}} \right)^* \boldsymbol{\Psi} \right)_i = \sum_{j \in N_i} |\boldsymbol{\alpha}_{ij}| \boldsymbol{\varepsilon}_{ij}^{(2)} (\boldsymbol{\Psi}_i - \boldsymbol{\Psi}_j) \Delta S_{ij} + N_i \sum_{j \in N_i} |\boldsymbol{\alpha}_{ij}| \cdot \boldsymbol{\varepsilon}_{ij}^{(4)} (\boldsymbol{\Psi}_j - \boldsymbol{\Psi}_i) \Delta S_{ij} - \sum_{j \in N_i, k \in N_j} |\boldsymbol{\alpha}_{kj}| \boldsymbol{\varepsilon}_{kj}^{(4)} (\boldsymbol{\Psi}_k - \boldsymbol{\Psi}_j) \Delta S_{kj} \quad (7)$$

式(7)并不是精确表达形式,而是采用了与文献[10]相同的近似方法,即人工耗散项式(3)中的耗散矩阵和耗散系数视为常数。此外本文还将拟拉普拉斯算子仍改为双调和算子以简化计算,数值实验表明,该简化方法对伴随解影响不大。

与流动方程式(1)的求解相同,对伴随方程式(6)采用5步Runge-Kutta时间推进求解,当地时间步长采用求解式(1)时最后一步的时间步长。

当前网格上的Euler流动方程和伴随方程求解完毕之后,可以获得该网格均匀加密后的新网格上的目标函数估算值。记 \mathbf{U}_h 和 $\boldsymbol{\Psi}_h$ 分别为细网格上流动解和伴随解, $\tilde{\mathbf{U}}_h$ 和 $\tilde{\boldsymbol{\Psi}}_h$ 分别为由粗网格插值到细网格上的流动解和伴随解,细网格上仅计及可计算误差项的目标函数 f_h 估算值为^[11]

$$f_h(\mathbf{U}_h) \approx f_h(\tilde{\mathbf{U}}_h) - \tilde{\boldsymbol{\Psi}}_h^T \mathbf{R}_h(\tilde{\mathbf{U}}_h) \quad (8)$$

2 网格自适应

2.1 自适应参数和自适应指示器

令 $\partial f_h = f_h(\tilde{\mathbf{U}}_h) - f_h(\mathbf{U}_h)$, 计及不可计算的剩余误差项,则有^[11]

$$\partial f_h \approx \tilde{\boldsymbol{\Psi}}_h^T \mathbf{R}_h(\tilde{\mathbf{U}}_h) + \mathbf{R}_h^p(\tilde{\boldsymbol{\Psi}}_h)^T (\mathbf{U}_h - \tilde{\mathbf{U}}_h) \quad (9)$$

$$\partial f_h \approx \tilde{\boldsymbol{\Psi}}_h^T \mathbf{R}_h(\tilde{\mathbf{U}}_h) + (\boldsymbol{\Psi}_h - \tilde{\boldsymbol{\Psi}}_h)^T \mathbf{R}_h(\tilde{\mathbf{U}}_h) \quad (10)$$

本文网格自适应的目标就是要减少式(9)和式(10)右端最后一项的两种形式的剩余误差项。应用中,为降低计算成本,并不直接在均匀加密网格上计算流动解 \mathbf{U}_h 和伴随解 $\boldsymbol{\Psi}_h$ 。而是以高、低阶插值之差来代替密网格上数值解与插值解之间的误差,即

$$\mathbf{U}_h - \tilde{\mathbf{U}}_h \approx \mathbf{U}^H - \mathbf{U}^L \quad (11)$$

$$\boldsymbol{\Psi}_h - \tilde{\boldsymbol{\Psi}}_h \approx \boldsymbol{\Psi}^H - \boldsymbol{\Psi}^L \quad (12)$$

式中: \mathbf{U}^H 和 \mathbf{U}^L 分别代表密网格上高、低阶的流动插值解, $\boldsymbol{\Psi}^H$ 和 $\boldsymbol{\Psi}^L$ 分别代表高、低阶的伴随插值解。这两种插值方法在下文详述。把用高、低阶插值解近似计算的两种形式的剩余误差的平均值设定为自适应参数,则网格单元 k 的自适应参数表示为

$$\varepsilon_k = \frac{1}{2} \sum_{l(k)} \{ | [\boldsymbol{\Psi}^H - \boldsymbol{\Psi}^L]_{l(k)}^T [\mathbf{R}_h(\mathbf{U}^L)]_{l(k)} | +$$

$$| [\mathbf{R}_h^p(\boldsymbol{\Psi}^L)]_{l(k)}^T [\mathbf{U}^H - \mathbf{U}^L]_{l(k)} | \} \quad (13)$$

方程(13)的总和是每个粗网格单元所包含的所有节点。

定义每个单元 k 的当地自适应指示器为

$$\eta_k = \frac{\varepsilon_k}{e_0} \quad (14)$$

式中: $\bar{e}_0 = e_0 / N_e$, e_0 为人为设定的全局误差阈值, N_e 为当前粗网格的单元数。如果 $\eta_k > 1$, 则该网格单元标记为需要加密。

令 $\varepsilon = \sum_k \varepsilon_k$ 代表当前网格上所有元素 ε_k 的总和,即目标函数的全局剩余误差。记 e_g 为人为设定的网格自适应加密终止的误差上限。当 $\varepsilon < e_g$ 时,网格加密终止。

将低阶的插值解代入式(8),可得细网格上目标函数的估算值,即目标函数修正值

$$f_h(\mathbf{U}_h) \approx f_h(\mathbf{U}^L) - (\boldsymbol{\Psi}^L)^T \mathbf{R}_h(\mathbf{U}^L) \quad (15)$$

2.2 网格加密方法

本文采用 h -加密方法^[12]来进行网格自适应。四面体单元的剖分方式分为常规加密和协调加密。常规加密即8分模式,将四面体分成8个子四面体;其几何质量与原四面体相近。协调加密是为了消除不同加密层级之间的悬点,有二分、四分、形心节点分割等不同的分割模式,剖分后获得的新四面体与原四面体不相似且质量有所下降。自适应网格加密时,被标记加密的单元采用常规加密,最后使用协调加密消除悬点。计算自适应参数时需要构造的全局加密网格则全部采用常规加密。

为保证自适应加密之后的新网格几何质量,还需进一步修正加密准则。每个四面体单元有4个面相邻的单元(外边界处除外)。对于某单元 k , 记自适应指示器式(14)大于1的相邻单元的个数为 m_k 。若 $\eta_k > 1$ 而 $m_k \leq 1$, 则标记单元 k 不需要加密;若 $\eta_k \leq 1$ 而 $m_k \geq 2$, 则标记单元 k 标记为需要加密。这样做实际上是光滑加密和非加密区域交界处的网格,消除了孤立的加密单元和孤立的非加密单元,减小协调加密的使用量,提高整体网格的几何质量。

对于加密后获得的新网格,位于物面上新添加的网格点通常不在物面上,需要将新的网格节点投影到物面上以保证网格边界与物面的贴合。针对数值实验中的M6机翼,本文使用B样条曲线拟合翼型离散点数据,在网格生成后对机翼上下表面所有节点进行调整。投影完成后,机翼表面曲率较大处获得的新网格单元的物面法向尺度往往小于另两个方向的网格尺度,多轮加密后往往会出现负体积网格,因此在每轮投影后需要调整与物面节点相邻的流场内部的网格节点。

2.3 插值方法

在计算目标函数估算值和自适应参数时,需将粗网格上的流动解和伴随解映射到相应的细网格,以近似获得细网格上的流动解和伴随解。这里需要用到两种插值解:低阶插值解和高阶插值解。对于节点格式,流动解和伴随解是储存在节点上的。以下给出解的高、低阶插值构造方法。

对于低阶插值,假设解是线性变化的。记任一条边的两端点分别为点 1 和点 2,加密后加入了新的网格节点 3,则节点 3 的低阶流动插值解为

$$U_3^L = (U_1 + U_2)/2 \quad (16)$$

式中 U_i 代表节点流动解或伴随解的任意分量。

对于高阶插值,则通过流动解和伴随解的值和梯度值的组合来构造高阶插值解。首先用格林-高斯方法求出每个节点处流动解与伴随解的梯度

$$\nabla U_i \approx \frac{1}{\Omega_i} \int_{\partial \Omega_i} U n dS \quad (17)$$

式中 Ω_i 为控制体 i 的体积。记 $\mathbf{l} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)^T$, 则节点 3 的高阶流动插值解为

$$U_3^H = (U_1 + U_2)/2 + (\nabla U_1 - \nabla U_2)^T \mathbf{l}/8 \quad (18)$$

式(18)相当于是在边的中点处利用端点数据得出的最小二乘拟合解。

3 数值实验

本文对 ONERA-M6 机翼的跨声速无黏绕流进行数值模拟,以验证本文所发展的误差估算与网格自适应方法的正确性和有效性。计算状态为 $Ma_\infty = 0.84$, $\alpha = 3.06^\circ$, 网格自适应处理的目标函数为阻力系数和升力系数。图 1 给出了机翼上表面的初始网格和相应的马赫数云图。

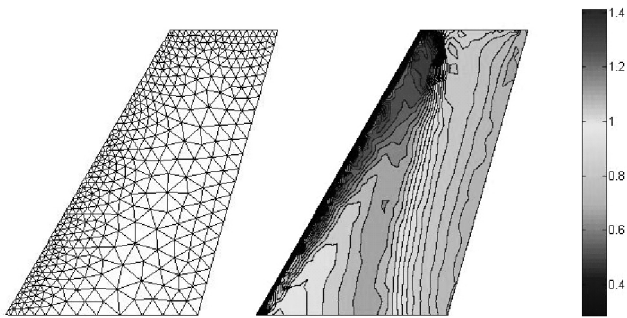


图 1 M6 机翼上表面初始网格和相应的马赫数云图

3.1 阻力系数

全局误差阈值 $e_0 = 0.03$, 网格自适应处理终止误差上限设为每轮粗网格阻力系数的 30%。全局误差阈值和终止误差上限是自定义的参数,全局误差阈值用于控制每轮

加密的网格量,而终止误差上限决定了自适应终止时的误差水平,可以说与最终一轮的网格节点数有关。初始网格节点总数为 11954。网格自适应加密 5 次后,达到了误差要求,最终网格节点数为 213967,图 2 展示了自适应处理后机翼上表面的网格和马赫数云图。可以看出,自适应后的网格在机翼前后缘和翼尖处进行了大量的加密;同时对 λ 激波处的加密也较为明显,但网格密度明显小于前后缘处的加密。

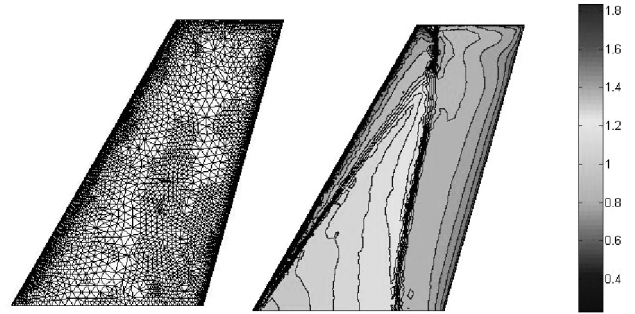


图 2 M6 机翼上表面自适应网格和相应的马赫数云图(目标函数为阻力系数)

图 3 给出了初始网格及两次均匀加密后的网格上计算获得的阻力系数、各层自适应网格的阻力系数以及修正值随网格节点数的变化曲线。图中亦给出了均匀网格结果的外插值(根据 Richardson 法获得)。从阻力变化曲线可以看出,通过网格自适应处理,阻力系数可以快速地趋近外推值,而修正处理进一步提高了计算结果的精确性。总之,应用本文方法可在节点数很小的自适应网格上获得准确的阻力系数值。

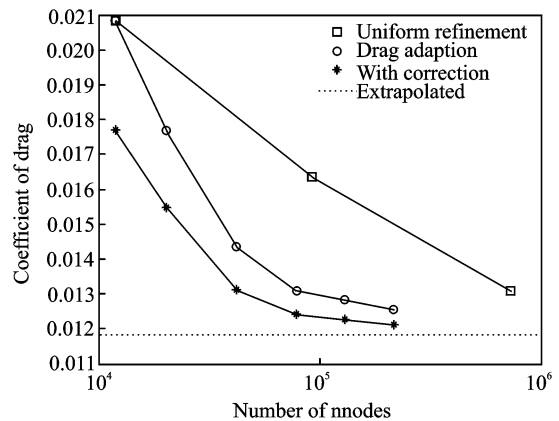


图 3 M6 机翼自适应阻力系数

3.2 升力系数

全局误差阈值 $e_0 = 0.4$, 终止误差上限设为每轮粗网格升力系数的 12%。在做升力自适应时,本文的取值控制了节点变化与阻力自适应是一致的,生成的最终的网格

也有数量相近的节点数,方便我们进行比较。

经过5轮自适应后,网格节点数量为221680,与阻力自适应的最终网格节点数差不多。图4给出了升力自适应后M6机翼上表面的网格分布和马赫数云图。比较图2和图4的机翼上表面网格,可以发现,升力自适应同样加密了机翼前后缘和翼尖处的网格。区别也是明显的,阻力自适应时, λ 激波处网格密度较低,靠近翼尖30%翼展的部分甚至并没有明显的加密;升力自适应更多地加密了 λ 激波处的网格,图4比图2的激波结构更清晰,这说明 λ 激波处网格的加密对升力系数的计算更重要。

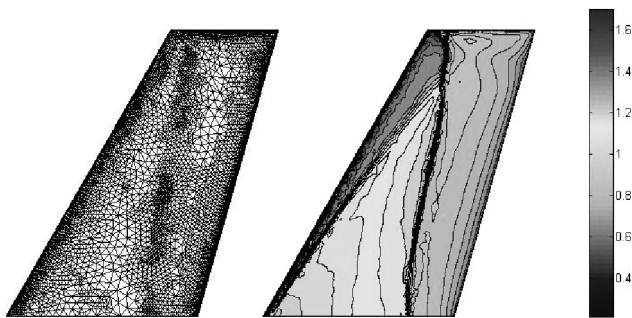


图4 M6机翼上表面自适应网格和相应的马赫数云图(目标函数为升力系数)

图5给出了初始网格及两次均匀加密后的网格上计算获得的升力系数、各层自适应网格的升力系数以及修正值随网格节点数的变化曲线,亦给出了均匀网格上升力系数的外插值。通过网格自适应处理,升力系数同样可以快速地趋近外推值,修正处理也提高了升力系数的精确性。

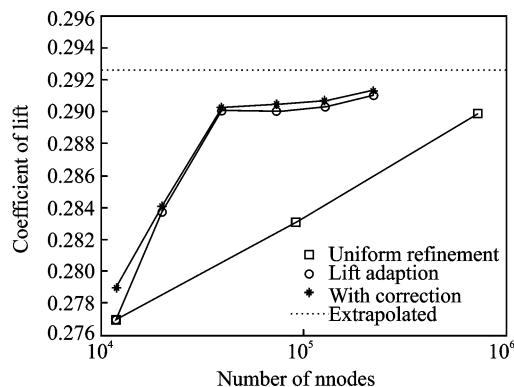


图5 M6机翼自适应升力系数

4 结束语

本文发展的基于伴随方程的三维四面体网格自适应处理能有效加密对目标函数更敏感的网格区域,最终获得的网格分布合理。数次自适应处理能使目标函数快速趋

近外插值,保证在网格节点数目较小的情况下获得准确的计算结果,有效地提高了目标函数的计算效率。通过对目标函数的修正,进一步提高了其计算精度。

参考文献

- [1] Pierce N A, Giles M B. Adjoint recovery of superconvergent functionals from PDE approximations [J]. SIAM Review, 2000, 42(2): 247-264.
- [2] Venditti D A, Darmofal D L. Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow [J]. Journal of Computational Physics, 2000, 164(1): 204-227.
- [3] Venditti D A, Darmofal D L. Grid adaptation for functional outputs: application to two-dimensional inviscid flows [J]. Journal of Computational Physics, 2002, 176(1): 40-69.
- [4] Venditti D A, Darmofal D L. Grid adaptation for functional outputs of 2-D compressible flow simulations [C]//Fluids 2000 Conference and Exhibit. 2002: 2244.
- [5] Venditti D A, Darmofal D L. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows [J]. Journal of Computational Physics, 2003, 187(1): 22-46.
- [6] Park M A. Adjoint-based, three-dimensional error prediction and grid adaptation [J]. AIAA Journal, 2004, 42(9): 1854-1862.
- [7] Jameson A, Schmidt W, Turkel E. Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes [C]//14th Fluid and Plasma Dynamics Conference. 1981: 1259.
- [8] Swanson R, Radespiel R, Turkel E, et al. Comparison of several dissipation algorithms for central difference schemes [C]//13th Computational Fluid Dynamics Conference. 1997: 1945.
- [9] Frink N. Recent progress toward a three-dimensional unstructured Navier-Stokes flow solver [C]//32nd Aerospace Sciences Meeting and Exhibit. 1994: 61.
- [10] Amoignon O. Adjoint-based aerodynamic shape optimization [D]. Uppsala Universitet, 2003.
- [11] Venditti D A, Darmofal D L. Grid adaptation for functional outputs: application to two-dimensional inviscid flows [J]. Journal of Computational Physics, 2002, 176(1): 40-69.
- [12] Löhner R, Baum J D. Adaptive h refinement on 3D unstructured grids for transient problems [J]. International Journal for Numerical Methods in Fluids, 1992, 14(12): 1407-1419.