wmake

SIMPLE PISO算法的评价

CFD难度 高级

1. 引言

本文讨论CFD中的单相可压缩模拟和不可压缩模拟。对于可压缩流动求解，分别存在连续性方程、动量方程以及能量方程，可以分别用于求解密度、速度、能量，并依托状态方程通过密度求解压力。对于不可压缩流动求解，由于不存在一个单独的压力方程，导致求解较为复杂。历史上，存在着各种不同的方法用于求解不可压缩流动，如步进法(Kim and Moin, 1985)、流函数法、任意可压缩法(Chorin, 1967)、投影法(Chorin, 1967)等。一些文献中认为步进法和投影法为同一种方法，即Kim and Moin于1985年提出的步进法为Chorin于1967年提出的投影法的变种(Hirsch, 2007)。SIMPLE(Patankar and Spalding, 1983)和PISO(Issa, 1986)均属于投影法。本文不涉及N-S方程离散等相关内容，有关N-S方程离散请参考icoFoam解析。

2. 可压缩流

首先看可压缩流的连续性方程、动量方程和能量方程：

$$\frac{\partial \rho}{\partial t}+\nabla\cdot\rho\mathbf{U}=0 \tag{1}$$

$$\frac{\partial \rho\mathbf{U}}{\partial t}+\nabla\cdot(\rho\mathbf{U}\mathbf{U})=-\nabla p+\nabla\cdot(\mu\nabla\mathbf{U}) \tag{2}$$

$$\frac{\partial \rho e}{\partial t}+\nabla\cdot(\rho\mathbf{U}e)+\frac{\partial \rho K}{\partial t}+\nabla\cdot(\rho\mathbf{U}K)-\nabla\cdot(\alpha_{\mathrm{eff}}\nabla e)=-\nabla\cdot(p\mathbf{U}) \tag{3}$$

对可压缩流动进行求解的时候，可采取以下方法：

return 0;

- 显式步进法：如Lax-wendroff、MacCormack法等；

- 耦合求解法：将速度压力能量耦合成一个大的线性系统同时求解；

wmake

- 分离求解法：详见下述；

其中的显式步进法以及耦合求解法均不需要SIMPLE或PISO循环。也就是说对于可压缩流动，可以通过下面的步骤求解：

1. 求解连续性方程获得密度，并通过状态方程更新压力；
2. 求解动量方程获得速度；
3. 求解能量方程获得温度；
4. 更新可压缩性，进入下一个时间步；

3. 不可压缩流

对于不可压缩流动，有不可压缩流体连续性方程和控制方程

$$\nabla \cdot \mathbf{U} = 0 \tag{4}$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \otimes \mathbf{U}) = -\nabla \frac{p}{\rho} + \nabla \cdot (\nu \nabla \mathbf{U}), \tag{5}$$

方程(4)、(5)具有下述特点：

- 方程(5)为一个关于速度的方程，可用来求解速度。不同于可压缩流动，方程(4)还是一个关于速度的方程，其中不包含压力。细看的话，其更像是一个限制性条件：速度场的散度要为零；
- 方程(5)的第二项为一个非线性项；
- 能量方程完全解耦，即表明不可压缩流动问题如果涉及到传热，可以在得到速度压力之后，对能量方程直接求解(Anderson, 2007)；
- 如果使用可压缩流动求解的方法，如Lax-wendroff显式法进行推进，为了稳定性需要满足时间步长的限制。这个时间步长和因素成反比，在不可压缩流动中，音速趋向于无穷大，因此导致稳定解的时间步长趋向于0(Anderson, 2007)；

return 0;

针对这些特性，不可压缩的求解通常采用投影法。投影法又进而分为显式投影法和隐式投影法。目前为了去除时间步长的限制，CFD中通常采用隐式投影法。例如，开源CFD软件OpenFOAM中植入的SIMPLE和PISO算法均属于隐式的投影法。在投影法中，

wmake

- 方程(5)还是一个关于速度的方程，同时把方程(4)转化为限制性条件，即转化为压力泊松方程；

$$\nabla\cdot(\mathbf{HbyA}^{n+1})=\nabla\cdot(\frac{1}{A_\mathrm{P}}\nabla p^{n+1}). \tag{6}$$

- 方程(5)的非线性项可选用非线性求解器，但CFD中则选择将其线性化，即转化为动量预测方程：

$$A_\mathrm{P}\mathbf{U}_\mathrm{P}^r+\sum A_\mathrm{N}\mathbf{U}_\mathrm{N}^n-E_\mathrm{P}=-\nabla p^n, \tag{7}$$

转化后的方程(6)和(7)的特点依然为速度$\mathbf{U}$依赖压力p，反之亦然。这个特点引致CFD中俩种算法：

- 耦合式算法：将方程(6)和(7)放置在一个大矩阵进行求解；
- 分离式算法：顺序求解方程(6)和(7)，最终满足收敛条件；

SIMPLE和PISO均属于分离式算法。PISO算法的思路为：

1. 首先求解方程(7)骤获得预测速度；
2. 通过预测速度求解方程(6)获得压力；
3. 通过压力修正速度，即：

$$\mathbf{U}_\mathrm{P}^{n+1}=\mathbf{HbyA}^{n+1}-\frac{1}{A_\mathrm{P}}\nabla p^{n+1}. \tag{8}$$

4. 回到第二步，直到收敛；

4. SIMPLE和PISO

return 0;

对于PISO算法的几点评价：

- 求解压力泊松方程之后对速度进行修正的来源来自于两个方面，即方程(8)右侧的第一项和第二项，第一项来自于相邻网格单元的速度，第二项来自于压力。

wmake

- 进一步的，方程(8)右侧的第一项$\mathbf{HbyA}$由离散后的矩阵系数A_P和相邻网格单元的速度$\mathbf{U}_\mathrm{N}$构成。在PISO中，修正压力后，会重新评估$\mathbf{HbyA}$中的$\mathbf{U}_\mathrm{N}$，但矩阵系数A_P保持不变。A_P不变的意义表明，其中的非线性项进行线性化后的影响被忽略。

对于SIMPLE算法的几点评价：

天呐、发出来之后发现写重复了，周末改一下

- ~~求解压力泊松方程之后对速度进行修正的来源来自于两个方面，即方程(8)右侧的第一项和第二项，第一项来自于相邻网格单元的速度，第二项来自于压力。~~

- ~~进一步的，方程(8)右侧的第一项$\mathbf{HbyA}$由离散后的矩阵系数A_P和相邻网格单元的速度$\mathbf{U}_\mathrm{N}$构成。在PISO中，修正压力后，会重新评估$\mathbf{HbyA}$中的$\mathbf{U}_\mathrm{N}$，但矩阵系数A_P保持不变。A_P不变的意义表明，其中的非线性项进行线性化后的影响被忽略。~~

最后，强调一点，速度压力耦合中的压力方程是整个求解过程中最关键的一步，整个求解器的效率依赖于它的表现。因此，所有的外挂如多重网格、矩阵预条件等黑科技、暴击都应该用于压力方程中，毫无保留。

参考文献

Kim, J., Moin, P., 1985. Application of a fractional-step method to incompressible Navier-Stokes equations. Journal of computational physics, 59, 308-323.

Chorin, A. J., 1967. The numerical solution of the Navier-Stokes equations for an incompressible fluid. Bulletin of the American Mathematical Society, 73, 928-931.

Chorin, A. J., 1967. A numerical method for solving incompressible viscous flow problems. Journal of computational physics, 2, 12-26.

Patankar, S. V., Spalding, D. B., 1983. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. In Numerical Prediction of Flow, Heat Transfer, Turbulence and Combustion, 13, 54-57

return 0;

Issa, R. I., 1986. Solution of the implicitly discretised fluid flow equations by operator-splitting. Journal of Computational Physics, 62, 40-65.

Hirsch, C., 2007. Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics. Butterworth-Heinemann.

Anderson, J. D., 2007. 计算流体力学基础及其应用.

wmake

*2018.05.24*大修

return 0;