

基于三维同位网格的高效稳定的分离式算法IDEAL

孙东亮, 屈治国, 何雅玲, 陶文铨*

西安交通大学动力工程多相流国家重点实验室, 西安 710049

* 联系人, E-mail: wqtao@mail.xjtu.edu.cn

2008-08-12 收稿, 2008-11-08 接受

国家自然科学基金重点项目(批准号: 50636050)、国家重点基础研究发展计划(编号: 2006CB601203; 2007CB206902)资助项目

摘要 分离式算法 IDEAL(inner doubly-iterative efficient algorithm for linked-equations)是一种高效稳定的算法, 在该算法中每个迭代层次上对压力方程进行两次内迭代计算, 第一次内迭代过程用于克服 SIMPLE 算法的第一个假设, 第二次内迭代过程用于克服 SIMPLE 算法的第二个假设. 这样在每个迭代层次上充分满足了速度和压力之间的耦合, 从而大大提高了计算的收敛速度和计算过程的稳定性. 把 IDEAL 算法推广到了三维同位网格系统, 其中界面流速计算采用修正的动量插值方法(MMIM), 这样计算得到的数值结果与亚松弛因子的选取无关. 最后通过 5 个不可压缩流动和传热的三维算例对 IDEAL 算法与其他三个被广泛使用的算法——SIMPLER, SIMPLEC 和 PISO 进行了比较. 通过分析比较得出 IDEAL 算法在收敛性和健壮性上均优于 SIMPLER, SIMPLEC 和 PISO 算法.

关键词

三维同位网格
分离式算法
IDEAL
内迭代
收敛性
健壮性

数值求解流动和传热问题的方法分为以压力为基本求解变量和以密度为基本求解变量两大类^[1,2]. 由于以压力为求解变量的方法可以同时适用于不可压缩及可压缩流动, 所以该方法得到了普遍的应用. 在以压力为求解变量的方法中, 压力修正算法是最普遍被使用的方法. 压力修正算法于 1972 年由 Patankar 和 Spalding^[3]提出, 并被命名为 SIMPLE 算法. SIMPLE 算法有两个基本假设: () 迭代初始压力和速度场单独进行设定, 压力和速度的内在联系没有得到体现; () 引入了压力修正, 并在压力修正方程的推导中, 忽略了邻点速度修正的影响. 以上两个假设不影响计算的最终结果^[4], 但会影响计算过程的稳定性和收敛性, 为了克服或者减轻这两点假设而引起的缺陷, 近三十年来提出了多种改进方案. 1981 年 Patankar 又提出了 SIMPLER 算法^[5], 该算法克服了 SIMPLE 算法任意给定压力初场的缺点, 压力的初值和改进值通过求解压力方程来得到, 而不是通过在原有的压力上加一个修正值来获得. 在 CSIMPLER^[6]中, 采用了相同的方法来克服 SIMPLE 算法的第一个假设, 但同时用原有的压力上加一个修正值作为迭

代求解改进值的初场. 1984 年 Van Doormaal 和 Raithby 提出了 SIMPLEC 算法^[7], 通过改变速度修正方程系数的定义, 部分地弥补了邻点速度修正的影响; 同时又提出了 SIMPLEX 算法^[8,9], 通过计算来确定压力修正方程的系数, 但是引入各点压力修正值之差相等的假定. PISO 算法^[10]对于压力修正提出两步或者多步修正来减轻忽略了邻点速度修正的影响. Yen 和 Liu^[11]提出显式修正步法, 使当前迭代层次速度显式地满足动量方程来加速收敛. 以上算法均采用压力修正值 p' 来改进压力和速度, 但是一旦引入了压力修正值概念, 为了获得可用于实际求解的压力修正方程, 在速度修正方程中不可避免地要忽略邻点速度修正的影响, 据此而构成的算法都是半隐的. 因此为了克服 SIMPLE 算法中的第二个假设, 需要直接改进压力, 而不应引进压力修正值, 以上思想是全隐算法 CLEAR^[12,13]的成功所在. 但是我们的进一步数值实验表明直接求解压力方程会削弱 CLEAR 算法的稳定性, 为了克服这一缺点, 文献[14]提出了一种将 CLEAR 与 SIMPLER 组合起来的方案, 并称之为 CLEARER, 它虽然在一定程度上能克服 CLEAR 健壮

性较差的缺点,但同时使得CLEARER又成为一个半隐的算法.本文作者在深入分析CLEAR算法的基础上提出了IDEAL算法(Inner doubly-iterative efficient algorithm for linked-equations)^[15,16].在IDEAL算法中每个迭代层次上对压力方程进行两次内迭代计算,第一次内迭代过程用于克服SIMPLE算法的第一个假设,第二次内迭代过程用于克服SIMPLE算法的第二个假设.这样在每个迭代层次上充分满足了速度和压力之间的耦合,从而大大提高了计算的收敛速度和计算过程的稳定性.

本文把IDEAL算法推广至三维同位网格系统,界面流速计算采用修正的动量插值方法(MMIM),这样计算得到的数值结果与亚松弛因子的选取无关.对于不规则的计算区域,采用了区域扩充方法^[17].最后本文通过5个不可压缩流动和传热的三维算例对IDEAL算法与其他三个被广泛使用的算法——SIMPLER, SIMPLEC和PISO进行了比较,验证了该算法的可行性.

1 界面流速的动量插值方法

在交错网格中,速度布置在控制容积的界面,在压力梯度离散过程中会自然引入相邻两点压差,保证了速度和压力之间的耦合.但是由于速度各个分量和其他标量变量交错储存,不是采用同一套控制容积,这对三维非结构化网格或非正交曲线坐标系计算程序的开发带来很多不便.同位网格的提出正是为了克服这一缺点,各个变量均置于同一套网格上,但由于压力梯度采用中心差分离散时无法引入相邻两点压差,不能分辨棋盘式压力场的产生,引起速度场和压力的失耦.由此可见,同位网格中的关键是解决速度场和压力场的失耦问题.1983年,Rhie和Chow^[18]在同位网格中提出采用动量插值(momentum interpolation method,简称MIM)计算界面速度来解决速度和压力的失耦问题,但MIM方法会导致收敛的解在一定程度上与亚松弛因子有关.为了解决这个问题,Kobayashi和Pereira^[19]提出在实施动量插值之前,使亚松弛因子还原为1,此法虽能消除亚松弛因子的影响但会降低算法的健壮性.综上所述,在同位网格中,要建立一个稳定且高效的算法必须解决以下三个方面的问题:()算法必须要避免不合理的棋盘式压力场分布;()收敛的解必须与亚松弛因子无关;()算法必须保证一定的健壮性.Majumdar提

出的修正的动量插值方法(MMIM)^[20]满足了以上要求,所以本文采用MMIM方法计算界面速度值.

在介绍MMIM方法之前,对控制方程及其离散过程进行简要介绍.图1为同位网格系统图,为了方便给出了二维的情况.控制方程如下所示:

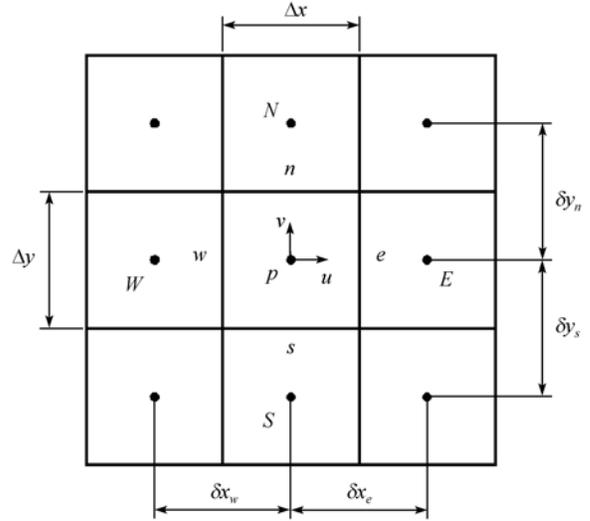


图1 同位网格系统图

连续性方程:

$$\frac{\partial(\rho u_f)}{\partial x} + \frac{\partial(\rho v_f)}{\partial y} + \frac{\partial(\rho w_f)}{\partial z} = 0, \quad (1)$$

动量方程:

$$\begin{aligned} & \frac{\partial(\rho u_f u)}{\partial x} + \frac{\partial(\rho v_f u)}{\partial y} + \frac{\partial(\rho w_f u)}{\partial z} \\ & = -\frac{\partial p}{\partial x} + \eta \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) + S_u, \end{aligned} \quad (2)$$

$$\begin{aligned} & \frac{\partial(\rho u_f v)}{\partial x} + \frac{\partial(\rho v_f v)}{\partial y} + \frac{\partial(\rho w_f v)}{\partial z} \\ & = -\frac{\partial p}{\partial y} + \eta \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) + S_v, \end{aligned} \quad (3)$$

$$\begin{aligned} & \frac{\partial(\rho u_f w)}{\partial x} + \frac{\partial(\rho v_f w)}{\partial y} + \frac{\partial(\rho w_f w)}{\partial z} \\ & = -\frac{\partial p}{\partial z} + \eta \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) + S_w, \end{aligned} \quad (4)$$

式中下标 f 表示界面.上述表达方式明确区分了作为求解变量的速度分量 u, v, w 和作为界面流速的 u_f, v_f, w_f 之间的区别,是研究同位网格的文献中经常采用的一种表达方法.

将亚松弛因子组织到迭代中的 u 方程离散形式为

$$\frac{a_p}{\alpha_u} u_p = \sum a_{nb} u_{nb} + b + A_p(p_w - p_e) + \frac{1 - \alpha_u}{\alpha_u} a_p u_p^0, \quad (5)$$

式中系数 a 和源项 b 的具体形式在文献[17]中有详细说明, α 为亚松弛因子.

方程(5)经变形可转化为

$$u_p = \alpha_u \left(\frac{\sum a_{nb} u_{nb} + b}{a_p} \right) + \frac{\alpha_u A_p}{a_p} (p_w - p_e) + (1 - \alpha_u) u_p^0, \quad (6)$$

令

$$\bar{u}_p = \frac{\sum a_{nb} u_{nb} + b}{a_p}, \quad (7)$$

$$d_p = \frac{\alpha_u A_p}{a_p}, \quad (8)$$

可得

$$u_p = \alpha_u \bar{u}_p + d_p (p_w - p_e) + (1 - \alpha_u) u_p^0, \quad (9)$$

其中界面上的压力通过线性差值由邻近的节点压力而得: $p_w = f_w^+ p_p + (1 - f_w^+) p_w$, $p_e = f_e^+ p_e + (1 - f_e^+) p_p$,

$$f_w^+ = \frac{\Delta x_w}{2\delta_w}, \quad f_e^+ = \frac{\Delta x_p}{2\delta_e}.$$

模仿方程(9), 可得界面处速度的表达式如下:

$$u_e = \alpha_u \bar{u}_e + d_e (p_p - p_E) + (1 - \alpha_u) u_e^0, \quad (10)$$

式中 \bar{u}_e, d_e 可通过相邻节点值线性插值得到:

$$\bar{u}_e = f_e^+ \bar{u}_E + (1 - f_e^+) \bar{u}_p, \quad (11)$$

$$d_e = f_e^+ d_E + (1 - f_e^+) d_p. \quad (12)$$

(10)式中亚松弛计算基于上一迭代层次计算所得的界面速度 u_e^0 , 而不是像 MIM 方法中亚松弛基于相邻节点速度线性插值得到的界面速度 $f_e^+ u_E^0 + (1 - f_e^+) u_p^0$, 这是 MMIM 与 MIM 的区别所在. 通过利用界面速度 u_e^0 可以消除亚松弛因子对计算结果的影响.

方程(10)可以重新写为

$$u_e = \tilde{u}_e + d_e (p_p - p_E), \quad (13)$$

其中假拟速度 \tilde{u}_e 为

$$\tilde{u}_e = \alpha_u \bar{u}_e + (1 - \alpha_u) u_e^0. \quad (14)$$

对于其他两个方向 v 和 w 的动量方程的离散形式以及界面速度的求解均与 u 方向的形式完全相同, 为了简便, 在本节中不再赘述, 在以后的叙述中我们使用 u 方向的相关方程作为三个方向方程的代表.

2 IDEAL 算法的计算步骤

IDEAL 算法的求解过程如图 2 所示. IDEAL 算法中在一个迭代层次上对压力方程进行了两次内迭代计算, 第一次内迭代过程用于克服 SIMPLE 算法的第一个假设, 第二次内迭代过程用于克服 SIMPLE 算法的第二个假设. IDEAL 算法在三维同位网格下的计算步骤如下:

步骤 1: 假设初始节点速度和界面流速分别为

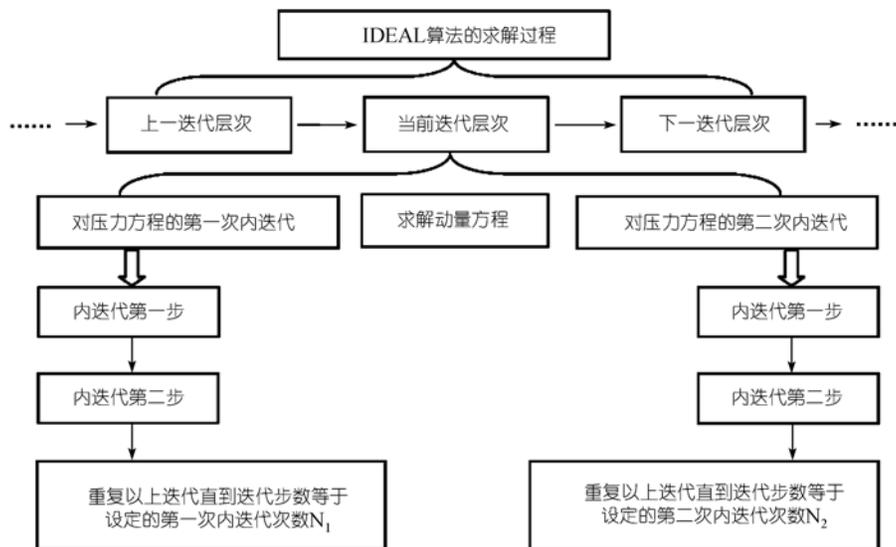


图 2 IDEAL 算法的求解过程

$$u_p^0, v_p^0, w_p^0, u_e^0, v_n^0, w_t^0.$$

步骤 2: 利用初始速度场计算 u, v, w 动量方程(5)的系数 a 和源项 b , 在每个迭代层次上动量方程的系数 a 和源项 b 保持不变.

-----对压力方程的第一次内迭代-----

步骤 3: 通过(14)式计算假拟速度 $\tilde{u}_e^0, \tilde{v}_n^0, \tilde{w}_t^0$. 假拟速度 \tilde{u}_e^0 的计算过程如下: 首先通过(7)式计算出 \bar{u}_p^0, \bar{u}_E^0 , 然后通过(11)式线性插值得出 \tilde{u}_e^0 , 最后通过(14)式计算得到假拟速度 \tilde{u}_e^0 , 其他界面的假拟速度求解方法相同.

步骤 4: 把求解界面速度 u_e, v_n, w_t 的方程(13)代入连续性方程(1)得到压力方程:

$$\begin{aligned} \frac{\alpha_p}{\alpha_p} p_p^{Temp} &= \sum a_{nb} p_{nb}^{Temp} + b, \\ a_p &= a_E + a_W + a_N + a_S + a_T + a_B, \\ a_E &= (\rho Ad)_e, a_W = (\rho Ad)_w, a_N = (\rho Ad)_n, \\ a_S &= (\rho Ad)_s, a_T = (\rho Ad)_t, a_B = (\rho Ad)_b, \\ b &= (\rho \tilde{u}^0 A)_w - (\rho \tilde{u}^0 A)_e + (\rho \tilde{v}^0 A)_s - (\rho \tilde{v}^0 A)_n \\ &\quad + (\rho \tilde{w}^0 A)_b - (\rho \tilde{w}^0 A)_t + (1 - \alpha_p) \frac{\alpha_p}{\alpha_p} p_p^{PTemp}, \end{aligned} \quad (15)$$

式中上标 $Temp$ 表示临时值, $PTemp$ 表示上一次内迭代步计算得到的临时值.

在第一次内迭代过程中可以对压力进行亚松弛处理, 这样可以使得计算过程更加稳定, 但是 IDEAL 算法本身具有较强的稳定性, 所以一般情况下压力不需要亚松弛.

步骤 5: 把解得的临时压力 p^{Temp} 代入(9)式计算出临时节点速度 $u_p^{Temp}, v_p^{Temp}, w_p^{Temp}$.

步骤 6: 将由步骤 4 和 5 计算出的临时节点速度和压力看作上一次内迭代步的临时节点速度和压力, 表示为 $u_p^{PTemp}, v_p^{PTemp}, w_p^{PTemp}$ 和 p^{PTemp} , 并返回步骤 3 开始下一次内迭代步, 同时步骤 3 和 4 内的上标为“0”的值用上标为“PTemp”的值代替. 重复以上迭代直到迭代步数等于已经设定的第一次内迭代次数 N_1 , 然后令第一次内迭代最终的压力作为初始压力 p^* . 通过对压力方程的第一次内迭代计算, 可以使得初始压力更加接近当前迭代层次上的最终压力, 更好地适应速度和压力的内在联系.

步骤 7: 根据求解出的压力 p^* 求解动量方程(5), 获得中间节点速度 u_p^*, v_p^*, w_p^* .

-----对压力方程的第二次内迭代-----

步骤 8: 通过(14)式计算假拟速度 $\tilde{u}_e^*, \tilde{v}_n^*, \tilde{w}_t^*$.

步骤 9: 把求解界面速度 u_e, v_n, w_t 的方程(13)代入连续性方程(1)得到压力方程:

$$\begin{aligned} \alpha_p p_p^{Temp} &= \sum a_{nb} p_{nb}^{Temp} + b, \\ a_p &= a_E + a_W + a_N + a_S + a_T + a_B, \\ a_E &= (\rho Ad)_e, a_W = (\rho Ad)_w, a_N = (\rho Ad)_n, \\ a_S &= (\rho Ad)_s, a_T = (\rho Ad)_t, a_B = (\rho Ad)_b, \\ b &= (\rho \tilde{u}^* A)_w - (\rho \tilde{u}^* A)_e + (\rho \tilde{v}^* A)_s \\ &\quad - (\rho \tilde{v}^* A)_n + (\rho \tilde{w}^* A)_b - (\rho \tilde{w}^* A)_t, \end{aligned} \quad (16)$$

在第二次内迭代过程中压力不需要亚松弛处理.

步骤 10: 把解得的临时压力 p^{Temp} 代入(9)式计算出临时节点速度 $u_p^{Temp}, v_p^{Temp}, w_p^{Temp}$.

步骤 11: 将由步骤 9 和 10 计算出的临时节点速度和压力看作上一次内迭代步的临时节点速度和压力, 表示为 $u_p^{PTemp}, v_p^{PTemp}, w_p^{PTemp}$ 和 p^{PTemp} , 并返回步骤 8 开始下一次内迭代步, 同时步骤 8 和 9 内的上标为“*”的值用上标为“PTemp”的值代替. 重复以上迭代直到迭代步数等于已经设定的第二次内迭代次数 N_2 , 然后令第二次内迭代最终的节点速度作为当前迭代层次上的最终节点速度 u_p, v_p, w_p . 最后把最终的节点速度和压力代入(13)式中得到当前迭代层次上最终的界面速度 u_e, v_n, w_t . 通过对压力方程的第二次内迭代计算, 可以使得当前迭代层次上的最终速度同时满足连续性方程和隐式的动量方程.

步骤 12: 求解其他标量方程.

步骤 13: 把当前迭代层次上的最终节点速度和界面速度看作下一个迭代层次上的初始节点速度和界面流速 $u_p^0, v_p^0, w_p^0, u_e^0, v_n^0, w_t^0$, 并返回步骤 2. 重复以上迭代直到获得收敛的解.

与 SIMPLER, CLEAR 一样 IDEAL 算法的初始压力 p^* 是通过求解压力方程获得的. 由于压力代数方程组是通过迭代方法求解的, 所以仍然需要一个压力初场, 并且压力初场的好坏对算法的收敛性和稳定性具有较大的影响, 在文献[6]中通过数值实例证实了这一观点. 在 IDEAL 算法中我们选用第一次内迭代所获得的压力 p^* 作为下一个迭代层次上的初始压力场, 我们的数值实验表明该初始压力场可以提高算法的收敛性和稳定性.

对压力方程的第二次内迭代计算像是第一次内

迭代的延续. 实际这两次内迭代具有本质上的区别, 第一次内迭代的目的是获得用于计算动量方程(5)的初始压力 p^* , 而第二次内迭代的目的是计算得出当前迭代层次上的最终速度.

在 IDEAL 算法中第一次内迭代次数 N_1 和第二次内迭代次数 N_2 (简称 $N_1 \& N_2$) 是可以调节的, 总的来说 $N_1 \& N_2$ 随着速度亚松弛因子的增大而增大, 因为速度亚松弛因子增大会导致计算过程不稳定, 严重时会导致计算过程发散, 而通常增加内迭代次数 $N_1 \& N_2$ 会提高计算过程的稳定性和收敛性.

3 算例比较条件和收敛标准

以下我们在三维同位网格情况下对 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在收敛性和健壮性方面进行系统的比较. 对四种算法进行比较的条件和收敛标准如下:

() 离散格式: 为了保证求解的精度和稳定性, 我们采用了 SGSD 格式^[21], 该格式至少二阶精度、绝对稳定. 同时采用了文献[22]中提出的延迟修正方法, 后来文献[23]为该方法提供了理论依据.

表 1 亚松弛因子 α 与时步倍率 E 的对照关系

| | | | | | | | | | |
|----------|-------|-----|-----|------|------|------|------|------|----------|
| α | 0.1 | 0.5 | 0.9 | 0.95 | 0.96 | 0.97 | 0.98 | 0.99 | 1 |
| E | 0.111 | 1 | 9 | 19 | 24 | 32.3 | 49 | 99 | ∞ |

() 代数方程组求解方法: 代数方程组采用交替方向线迭代方法(ADI)求解.

() 亚松弛因子: 在 SIMPLER 和 IDEAL 算法中压力亚松弛因子取为 1, 对于 SIMPLEC 和 PISO 算法压力不需要亚松弛处理. 在对以上四种算法进行比较的过程中三个方向的速度及温度亚松弛因子采用相同的值. 为了更好地比较各算法的健壮性, 引入了时步倍率 E , 其定义为

$$E = \frac{\alpha}{1-\alpha} \quad (0 < \alpha < 1). \quad (17)$$

通过引入时步倍率可以在更大范围内比较算法, 亚松弛因子和时步倍率的对照关系列于表 1 中.

() 网格系统: 采用均匀网格系统.

() 收敛标准: 质量和动量相对最大残差均小于某一小量^[17], 有关具体数值在各例中给出.

4 数值算例比较

以下通过 5 个三维流动与传热问题算例对 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法进行系统的

比较分析: () 方腔顶盖驱动流动; () 复杂方腔顶盖驱动流动; () 外掠后台阶流动; () 复杂通道内流动; () 方腔自然对流.

算例 1 至算例 4 属于强制对流问题, 其中算例 1, 2 属于闭口系统, 算例 3, 4 属于开口系统. 算例 5 属于速度和温度耦合的自然对流问题, 采用了 Boussinesq 假设^[24]. 对以上问题的计算基于以下条件: 层流、不可压缩、稳态、常物性.

4.1 算例 1: 方腔顶盖驱动流动

在三维问题中, 方腔顶盖驱动流动的数值基准解^[25-27]经常用来考核程序和比较算法. 方腔顶盖驱动流动的结构如图 3 所示. 计算在 $Re=100 \sim 1000$, 网格数= $32 \times 32 \times 32 \sim 82 \times 82 \times 82$ 的范围内进行, 判断收敛标准是质量和 u, v, w 动量残差均小于 10^{-8} . 雷诺数定义为

$$Re = \frac{u_{lid} H}{\nu}. \quad (18)$$

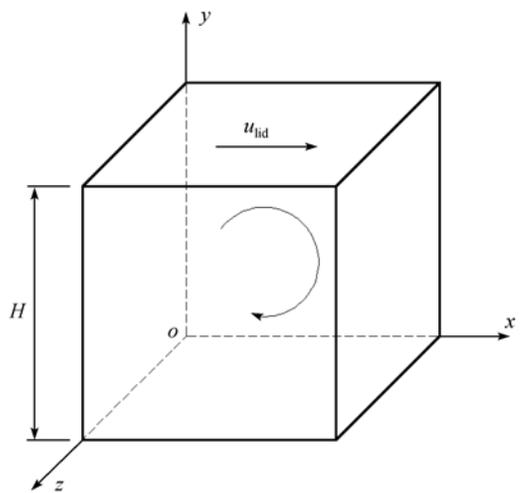


图 3 方腔顶盖驱动流动的结构图(算例 1)

图 4 显示了 $z=0.5H$ 截面中心线上的速度分布, 从图中可以看出 IDEAL 算法计算所得的结果与文献[27]中提供的数值基准解相一致, 说明 IDEAL 算法的可行性. 在图 5~7 中在不同网格数和不同雷诺数下对 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面进行了比较. IDEAL 算法的内迭代次数 $N_1 \& N_2$ 显示在图的上方, 例如图 5(a) 中在时步倍率 E 的两个区间范围内 $N_1 \& N_2$ 分别是 1&1 和 1&2. 通过图 5~7 可以发现在 IDEAL 算法中随着时步倍率, 即亚松弛因子的增加内迭代次数

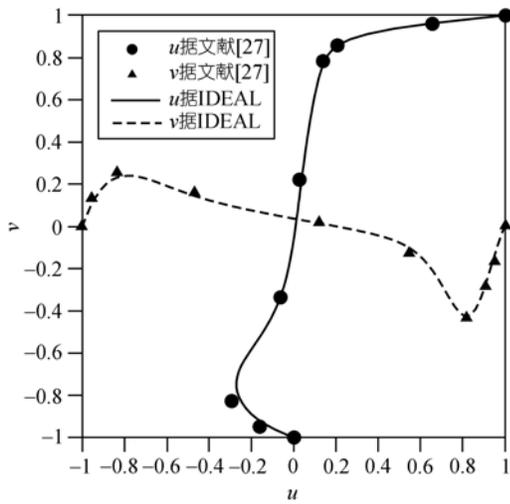


图4 $Re = 1000$ 时 $z = 0.5H$ 截面中心线上的速度分布比较(算例1)

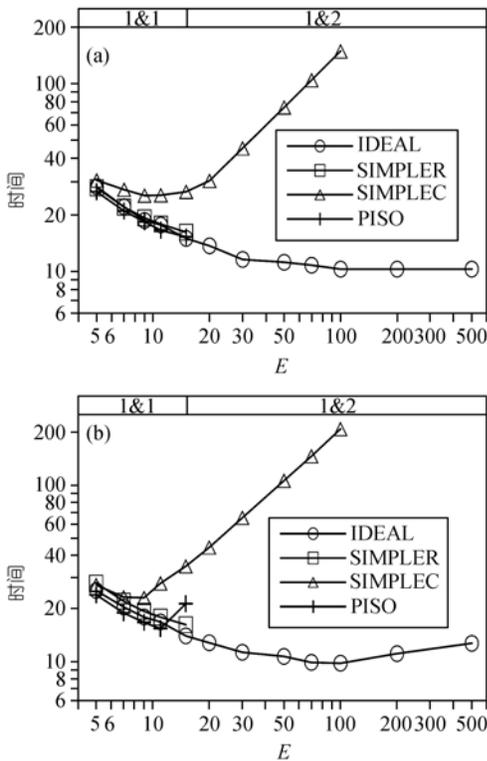


图5 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面的比较

算例1, 网格数=32×32×32. (a) $Re=100$, (b) $Re=300$

N_1 & N_2 也逐渐增加; IDEAL 算法的健壮性远远好于 SIMPLER, SIMPLEC 和 PISO 算法, 算例1中该算法几乎可以在任意时步倍率下收敛; 在计算时间方面

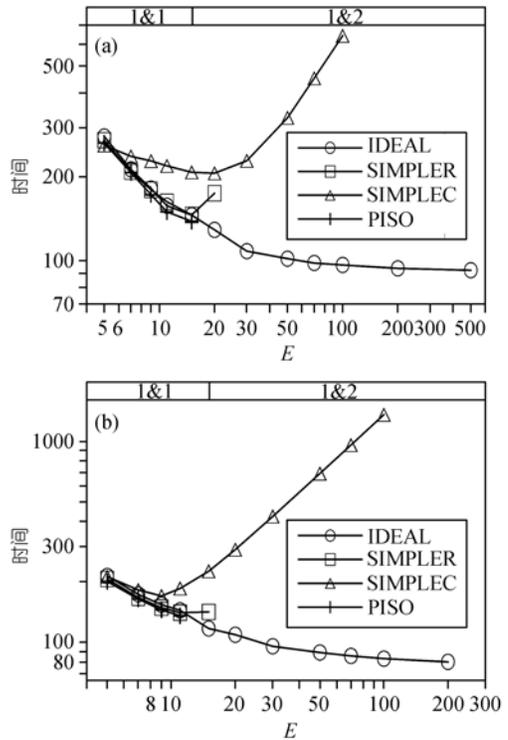


图6 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面的比较

算例1, 网格数=52×52×52. (a) $Re=100$, (b) $Re=500$

SIMPLEC 花费的计算时间最长, 甚至在网格数 $82 \times 82 \times 82$ 情况下 SIMPLEC 算法很难获得收敛的解, 所以在图7中没有显示 SIMPLEC 算法, 其次是 SIMPLER 和 PISO 算法, IDEAL 算法花费的计算时间最短.

表2显示了在算例1的不同情况下 IDEAL 算法所需最短计算时间较 SIMPLER, SIMPLEC 或 PISO 算法所需最短计算时间的减少比率, IDEAL 算法较 SIMPLER 算法减少的计算时间为 33.8%~46.6%; IDEAL 算法较 SIMPLEC 算法减少的计算时间为 55.1%~59.4%, 并且在网格数 $82 \times 82 \times 82$ 情况下 SIMPLEC 算法很难获得收敛的解; IDEAL 算法较 PISO 算法减少的计算时间为 32.7%~43.4%.

4.2 算例2: 复杂方腔顶盖驱动流动

算例1属于简单闭口系统, 通过算例1得出在简单闭口系统中 IDEAL 算法在健壮性和收敛性方面皆优于 SIMPLER, SIMPLEC 和 PISO 算法. 为了证实复杂闭口系统中 IDEAL 算法的性能仍然优于其他三种算法, 我们构造了算例2的流动结构, 如图8所示,

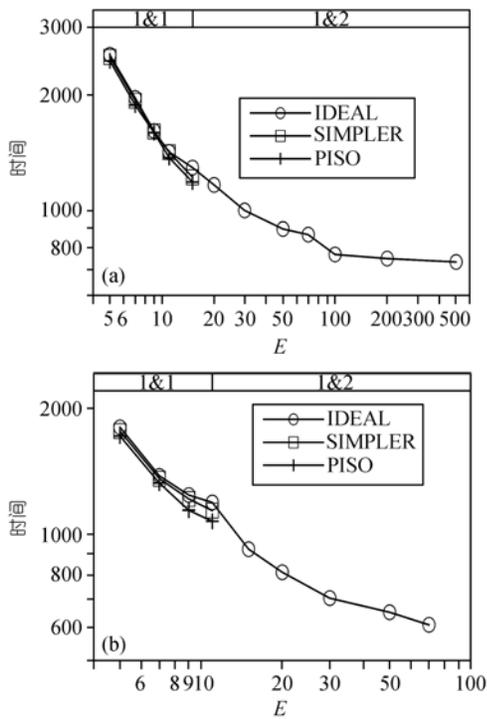


图7 IDEAL 算法与 SIMPLER 和 PISO 算法在计算时间和健壮性方面的比较

算例1, 网格数=82×82×82. (a) Re=100, (b) Re=1000

表2 IDEAL 算法所需最短计算时间较 SIMPLER, SIMPLEC 或 PISO 算法所需最短计算时间的减少比率(算例1)(%)

| 网格数 | 32×32×32 | | 52×52×52 | | 82×82×82 | |
|-----------------|----------|------|----------|------|----------|------|
| Re | 100 | 300 | 100 | 500 | 100 | 1000 |
| 相对 SIMPLER 减小比率 | 36.4 | 33.8 | 36.7 | 43.0 | 39.8 | 46.6 |
| 相对 SIMPLEC 减小比率 | 59.4 | 57.4 | 55.1 | 52.9 | - | - |
| 相对 PISO 减小比率 | 32.7 | 36.4 | 33.2 | 41.0 | 38.1 | 43.4 |

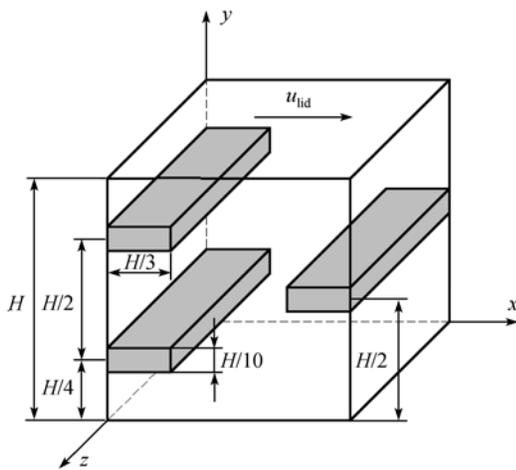


图8 复杂方腔顶盖驱动流动的结构图(算例2)

计算在 $Re = 100\sim 800$, 网格数=52×52×52~82×82×82 的范围内进行, 判断收敛标准是质量和 u, v, w 动量残差均小于 10^{-8} . 雷诺数定义为

$$Re = \frac{u_{lid}H}{\nu} \quad (19)$$

图9显示了 $z = 0.5H$ 截面沿 y 轴方向中心线上的 u 速度分布, 从图中可以看出 IDEAL 算法计算所得的结果与其他三种算法的计算结果相一致, 说明 IDEAL 算法的可行性. 在图10和11中在不同网格数和不同雷诺数下对 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面进行了比较. 从图中可以看出与简单闭口系统相同, 在复杂闭口系统中 IDEAL 算法仍旧保持其较好的收敛性和健壮性.

表3显示了在算例2的不同情况下 IDEAL 算法所需最短计算时间较 SIMPLER, SIMPLEC 或 PISO 算法所需最短计算时间的减少比率, IDEAL 算法较 SIMPLER 算法减少的计算时间为 29.0%~40.6%; IDEAL 算法较 SIMPLEC 算法减少的计算时间为 52.5%~61.4%; IDEAL 算法较 PISO 算法减少的计算

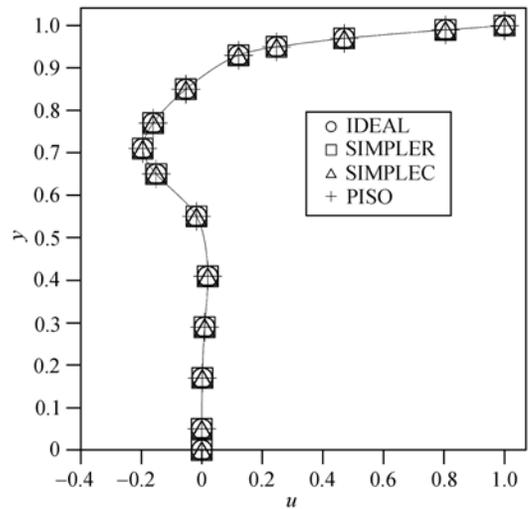


图9 Re=500 时 $z=0.5H$ 截面中心线上的 u 速度分布比较 (算例2)

表3 IDEAL 算法所需最短计算时间较 SIMPLER, SIMPLEC 或 PISO 算法所需最短计算时间的减少比率(算例2)(%)

| 网格数 | 52×52×52 | | 82×82×82 | |
|-----------------|----------|------|----------|------|
| Re | 100 | 500 | 100 | 800 |
| 相对 SIMPLER 减小比率 | 29.6 | 34.9 | 29.0 | 40.6 |
| 相对 SIMPLEC 减小比率 | 52.5 | 61.4 | 53.6 | 57.6 |
| 相对 PISO 减小比率 | 27.5 | 30.5 | 27.1 | 37.8 |

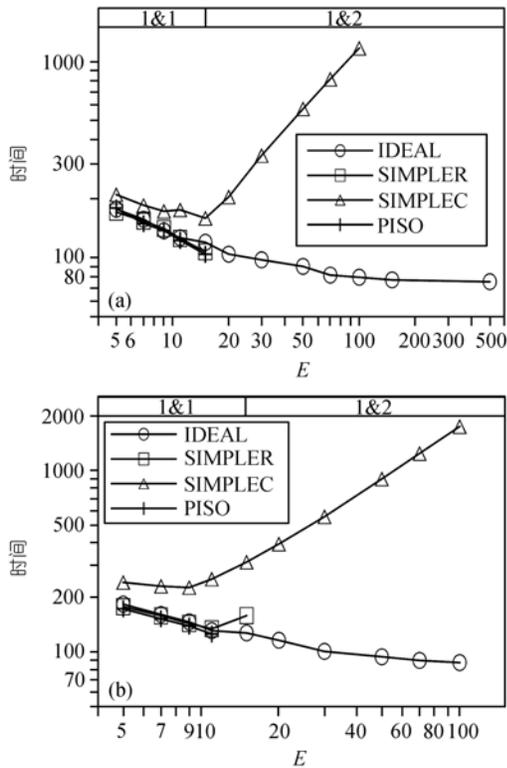


图 10 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面的比较

算例 2, 网格数=52×52×52. (a) Re=100, (b) Re=500

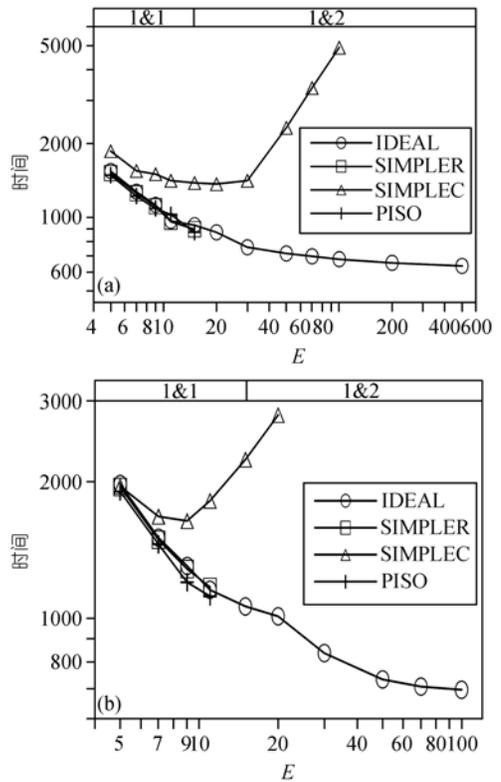


图 11 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面的比较

算例 2, 网格数=82×82×82. (a) Re=100 (b) Re=800

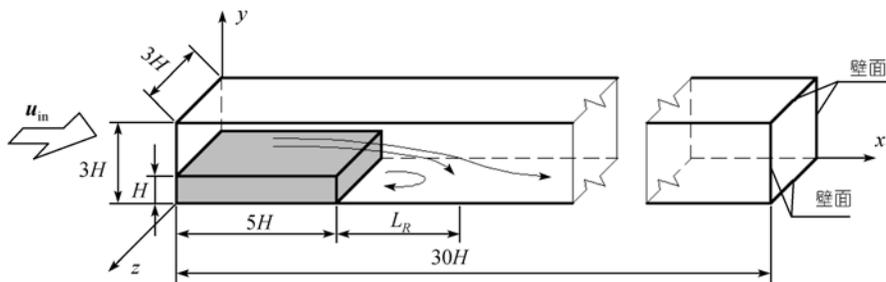


图 12 外掠后台阶流动的结构图(算例 3)

时间为 27.1%~37.8%.

4.3 算例 3: 外掠后台阶流动

外掠后台阶流动属于简单开口系统, 流动结构如图 12 所示. 计算在 $Re = 100 \sim 300$, 网格数= $80 \times 20 \times 20 \sim 160 \times 41 \times 41$ 的范围内进行, 判断收敛标准是质量和 u, v, w 动量残差均小于 10^{-7} . 雷诺数定义为

$$Re = \frac{u_{in} H}{\nu} \quad (20)$$

在图 13 和 14 中在不同网格数和不同雷诺数下对 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面进行了比较. 如图中所示, 在算例 3 中 SIMPLER 算法的健壮性最差, 其次是 SIMPLEC 和 PISO 算法, IDEAL 算法的健壮性最好, 几乎可以在任意时步倍率下收敛. 在计算时间方面 SIMPLEC 花费的计算时间最长, 其次是 SIMPLER 和 PISO 算法, IDEAL 算法花费的计算时间最短.

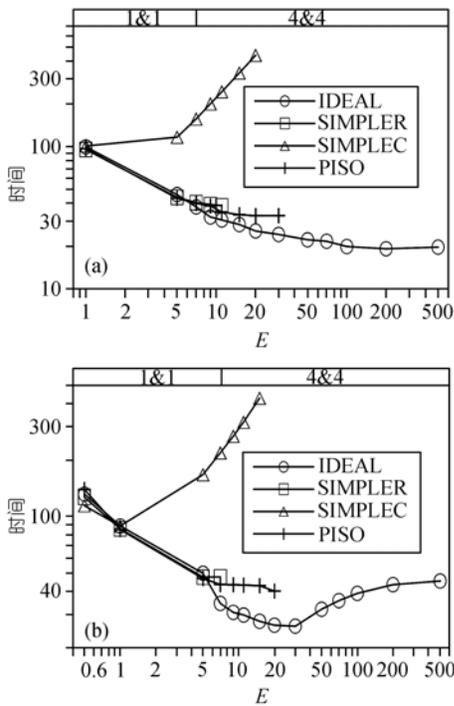


图13 IDEAL算法与SIMPLER, SIMPLEC和PISO算法在计算时间和健壮性方面的比较
算例3, 网格数=80×20×20. (a) Re=100, (b) Re=300

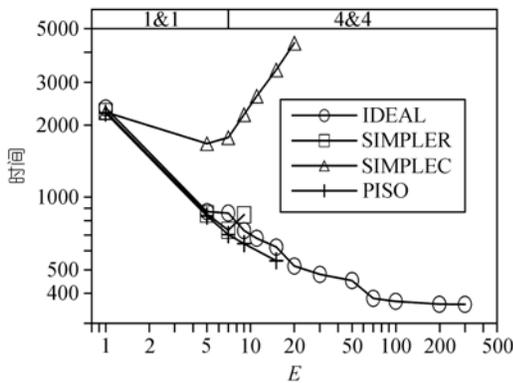


图14 IDEAL算法与SIMPLER, SIMPLEC和PISO算法在计算时间和健壮性方面的比较
算例3, 网格数=160×41×41, Re=100

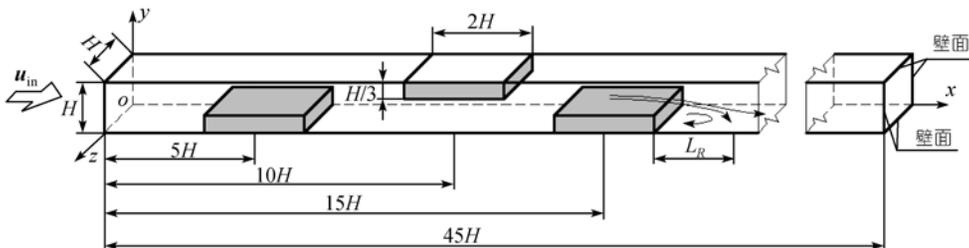


图15 复杂通道内的流动结构图(算例4)

表4显示了在算例3的不同情况下 IDEAL 算法所需最短计算时间较 SIMPLER, SIMPLEC 或 PISO 算法所需最短计算时间的减少比率, IDEAL 算法较 SIMPLER 算法减少的计算时间为 45.2%~50.7%; IDEAL 算法较 SIMPLEC 算法减少的计算时间为 70.7%~81.1%; IDEAL 算法较 PISO 算法减少的计算时间为 34.0%~41.6%.

表4 IDEAL 算法所需最短计算时间较 SIMPLER, SIMPLEC 或 PISO 算法所需最短计算时间的减少比率(算例3)(%)

| 网格数 | 80×20×20 | | 160×41×41 |
|-----------------|----------|------|-----------|
| Re | 100 | 300 | 100 |
| 相对 SIMPLER 减小比率 | 50.1 | 45.2 | 50.7 |
| 相对 SIMPLEC 减小比率 | 81.1 | 70.7 | 78.5 |
| 相对 PISO 减小比率 | 41.6 | 34.9 | 34.0 |

4.4 算例4: 复杂通道内流动

复杂通道内的流动结构如图15所示, 为了使通道内形成复杂的流动, 三块挡板被放置在通道内, 该结构属于复杂开口系统. 该算例的提出是为了验证在复杂开口系统中 IDEAL 算法是否仍能优于其他三种算法. 计算在 $Re = 100 \sim 500$, 网格数 = $150 \times 20 \times 20 \sim 190 \times 29 \times 29$ 的范围内进行, 判断收敛标准是质量和 u, v, w 动量残差均小于 10^{-7} . 雷诺数定义为

$$Re = \frac{u_{in} H}{\nu} \quad (21)$$

图16和17在不同网格数和不同雷诺数下对 IDEAL 算法与 SIMPLER、SIMPLEC 和 PISO 算法在计算时间和健壮性方面进行了比较. 从图中可以看出在复杂开口系统中 IDEAL 算法在健壮性和收敛性上仍旧优于 SIMPLER, SIMPLEC 和 PISO 算法.

表5显示了在算例4的不同情况下 IDEAL 算法所需最短计算时间较 SIMPLER, SIMPLEC 或 PISO 算法所需最短计算时间的减少比率, IDEAL 算法较 SIMPLER 算法减少的计算时间为 46.0%~53.7%; IDEAL 算法较 SIMPLEC 算法减少的计算时间为 56.4%~69.3%; IDEAL 算法较 PISO 算法减少的计算

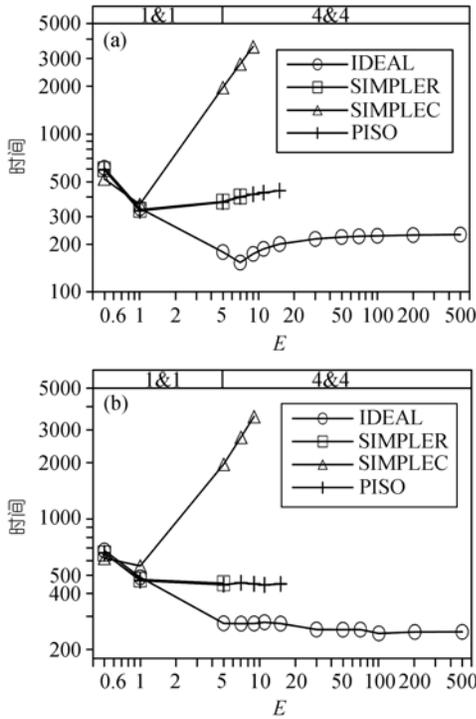


图 16 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面的比较

算例 4, 网格数=150×20×20. (a) $Re=100$, (b) $Re=300$

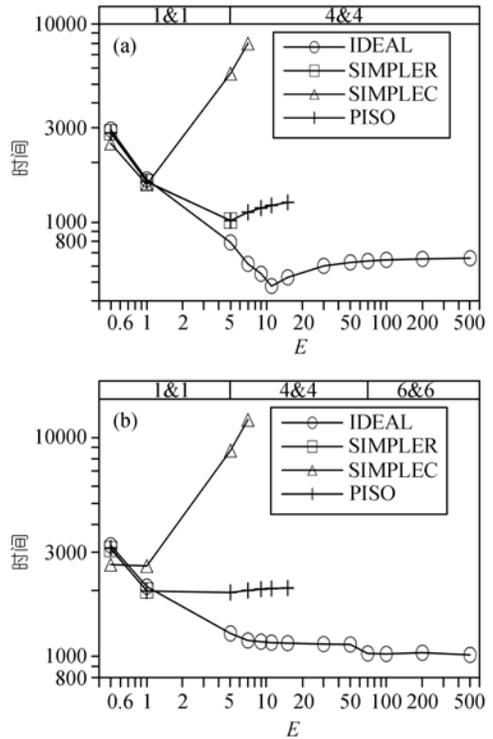


图 17 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面的比较

算例 4, 网格数=190×29×29. (a) $Re=100$, (b) $Re=500$

表 5 IDEAL 算法所需最短计算时间较 SIMPLER, SIMPLEC 或 PISO 算法所需最短计算时间的减少比率(算例 4)(%)

| 网格数 | 150×20×20 | | 190×29×29 | |
|-----------------|-----------|------|-----------|------|
| Re | 100 | 300 | 100 | 500 |
| 相对 SIMPLER 减小比率 | 53.7 | 46.0 | 53.3 | 49.2 |
| 相对 SIMPLEC 减小比率 | 57.2 | 56.4 | 69.3 | 60.8 |
| 相对 PISO 减小比率 | 52.9 | 44.9 | 53.2 | 48.1 |

时间为 44.9%~53.2%.

4.5 算例 5: 方腔自然对流

方腔自然对流是典型的速度和温度耦合问题, 流动结构如图 18 所示, 方腔有四个绝热面, 两个相对的不同温度的等温面. 计算在 $Ra = 10^4 \sim 10^5$, 网格数=30×30×30~50×50×50 的范围内进行, 判断收敛标准是质量和 u, v, w 动量残差均小于 10^{-7} . 瑞利数定义为

$$Ra = \frac{\rho g \beta H^3 (T_H - T_C)}{\alpha \mu} \quad (22)$$

表 6 显示了 IDEAL 算法计算出的壁面平均努塞数与文献[28,29]中给出的计算结果. IDEAL 算法的

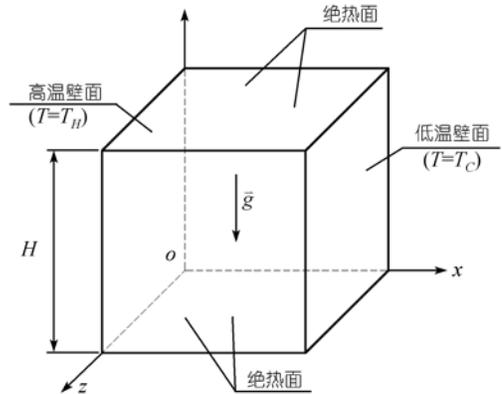


图 18 方腔自然对流结构图(算例 5)

计算结果与 Fusegi 等人^[28]和 Wakashima 与 Saitoh^[29]计算出的结果相一致. 图 19 为 $Ra=10^4$ 时 $z=0.5H$ 截面上 IDEAL 算法计算得出的压力场, 从图中可以看出修正的动量插值方法(MMIM)能有效消除棋盘式的压力分布. 图 20 和 21 在不同网格数和不同瑞利数下对 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面进行了比较. 从图中可以

表6 IDEAL 算法所计算的壁面平均努塞尔数与文献结果对比(算例5)

| Ra | 10^4 | 10^5 |
|-----------------------------|--------|--------|
| Fusegi等人 ^[28] | 2.1000 | 4.3610 |
| Wakashima等人 ^[29] | 2.0814 | 4.4309 |
| IDEAL | 2.0773 | 4.4016 |

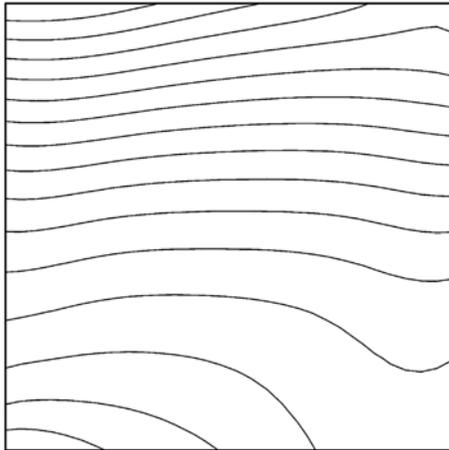


图19 $Ra = 10^4$ 时 $z = 0.5H$ 截面上 IDEAL 算法计算得出的压力场(算例5)

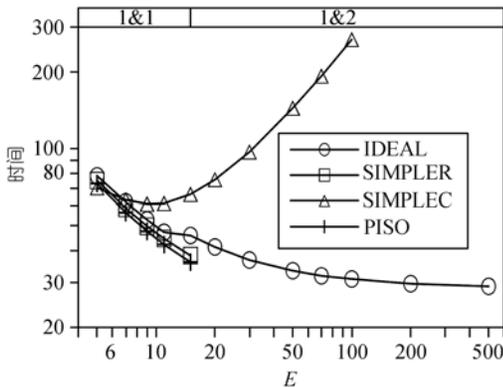


图20 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面的比较
算例5, 网格数=30×30×30, $Ra=10^4$

看出, 与流动问题一样在速度和温度耦合问题上 IDEAL 算法仍然保持其优越特性, 在健壮性和收敛性方面优于其他三种算法。

表7显示了在算例5的不同情况下 IDEAL 算法所需最短计算时间较 SIMPLER, SIMPLEC 或 PISO 算法所需最短计算时间的减少比率, IDEAL 算法较 SIMPLER 算法减少的计算时间为 16.3%~24.3%;

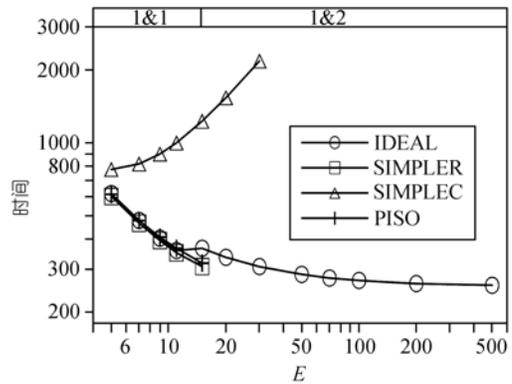


图21 IDEAL 算法与 SIMPLER, SIMPLEC 和 PISO 算法在计算时间和健壮性方面的比较
算例5, 网格数=50×50×50, $Ra=10^5$

表7 IDEAL 算法所需最短计算时间较 SIMPLER, SIMPLEC 或 PISO 算法所需最短计算时间的减少比率(算例5)(%)

| 网格数 | 30×30×30 | 50×50×50 |
|-----------------|----------|----------|
| Re | 10^4 | 10^5 |
| 相对 SIMPLER 减小比率 | 24.3 | 16.3 |
| 相对 SIMPLEC 减小比率 | 52.4 | 66.8 |
| 相对 PISO 减小比率 | 19.7 | 19.0 |

IDEAL 算法较 SIMPLEC 算法减少的计算时间为 52.4%~66.8%; IDEAL 算法较 PISO 算法减少的计算时间为 19.0%~19.7%。

5 结论

本文把 IDEAL 算法推广至三维同位网格系统, 并对 IDEAL 算法与最广泛使用的三种算法——SIMPLER, SIMPLEC 和 PISO 进行了系统的比较分析, 得出以下结论:

() 在三维同位网格上 IDEAL 算法在健壮性和收敛性方面皆优于其他三种算法;

() 在我们研究的5个算例中, IDEAL 算法几乎可以在任意时步倍率, 即亚松弛因子下得到快速收敛的解;

() IDEAL 算法所需最短计算时间较 SIMPLER, SIMPLEC 或 PISO 算法所需最短计算时间的减少比率分别为: IDEAL 算法较 SIMPLER 算法减少比率为 16.3%~53.7%; IDEAL 算法较 SIMPLEC 算法减少比率为 52.4%~81.1%; IDEAL 算法较 PISO 算法减少比率为 19.0%~53.2%。

参考文献

- 1 Shyy W, Mittal R. Solution methods for the incompressible Navier-Stokes equations. In: Johnson R W, ed. *The Handbook of Fluid Dynamics*. Boca Raton: CRC Press, 1998. 31.1—31.33
- 2 陶文铨. *计算传热学近代进展*. 北京: 科学出版社, 2000
- 3 Patankar S V, Spalding B. A calculation procedure for heat mass and momentum transfer in three dimensional parabolic flows. *Int J Heat Mass Transfer*, 1972, 15: 1787—1806
- 4 Patankar S V. *Numerical heat transfer and fluid flow*. Washington, DC: Hemisphere, 1980
- 5 Patankar S V. A calculation procedure for two-dimensional elliptic situations. *Numer Heat Transfer*, 1981, 4: 409—425[DOI]
- 6 Liu X L, Tao W Q, He Y L. A simple method for improving the SIMPLER algorithm for numerical simulations of incompressible fluid flow and heat transfer problems. *Eng Comput*, 2005, 22: 921—939[DOI]
- 7 Van Doormaal J P, Raithby G D. Enhancements of the SIMPLE method for predicting incompressible fluid flows. *Numer Heat Transfer*, 1984, 7: 147—163[DOI]
- 8 Van Doormaal J P, Raithby G D. An evaluation of the segregated approach for predicting incompressible fluid flow. *ASME Paper 85-HT-9*, 1985
- 9 Raithby G D, Schneider G E. Elliptic system: finite difference method II. In: Minkowycz W J, Sparrow E M, Pletcher R H, et al. eds. *Handbook of Numerical Heat Transfer*. New York: Wiley, 1988. 241—289
- 10 Issa R I. Solution of implicitly discretized fluid flow equation by operator-splitting. *J Comput Phys*, 1985, 62: 40—65[DOI]
- 11 Yen R H, Liu C H. Enhancement of the SIMPLE algorithm by an additional explicit corrector step. *Numer Heat Transfer, Part B*, 1993, 24: 127—141[DOI]
- 12 Tao W Q, Qu Z G, He Y L. A novel segregated algorithm for incompressible fluid and heat transfer problems-CLEAR (coupled and linked equations algorithm revised) Part I: Mathematical formulation and solution procedure. *Numer Heat Transfer, Part B*, 2004, 45: 1—17
- 13 Tao W Q, Qu Z G, He Y L. A novel segregated algorithm for incompressible fluid and heat transfer problems-CLEAR (coupled and linked equations algorithm revised) Part II: Application examples. *Numer Heat Transfer, Part B*, 2004, 45: 19—48
- 14 Cheng Y P, Lee T S, Low H T, et al. An efficient and robust numerical scheme for the SIMPLER algorithm on non-orthogonal curvilinear coordinates: CLEARER. *Numer Heat Transfer, Part B*, 2007, 51: 433—461[DOI]
- 15 Sun D L, Qu Z G, He Y L, et al. An efficient segregated algorithm for incompressible fluid flow and heat transfer problems-IDEAL (inner doubly-iterative efficient algorithm for linked-equations) Part I: Mathematical formulation and solution procedure. *Numer Heat Transfer, Part B*, 2008, 53: 1—17[DOI]
- 16 Sun D L, Qu Z G, He Y L, et al. An efficient segregated algorithm for incompressible fluid flow and heat transfer problems-IDEAL (inner doubly-iterative efficient algorithm for linked-equations) Part II: Application examples. *Numer Heat Transfer, Part B*, 2008, 53: 18—38[DOI]
- 17 陶文铨. *数值传热学*, 第2版. 西安: 西安交通大学出版社, 2001
- 18 Rhie C M, Chow W L. Numerical study of the turbulent flow past an airfoil with trailing edge separations. *AIAA J*, 1983, 21(11): 1525—1535[DOI]
- 19 Kobayashi M H, Pereira J C F. Numerical comparison of momentum interpolation methods and pressure-velocity algorithm using non-staggered grids. *Commun Appl Numer Meth*, 1991, 7: 173—196[DOI]
- 20 Majumdar S. Roles of under-relaxation in momentum interpolation for calculation of flow with non-staggered grids. *Numer Heat Transfer*, 1988, 13: 125—132[DOI]
- 21 Li Z Y, Tao W Q. A new stability-guaranteed second-order difference scheme. *Numer Heat Transfer, Part B*, 2002, 42: 349—365[DOI]
- 22 Khosla P K, Rubin S G. A diagonally dominant second order accurate implicit scheme. *Comput Fluids*, 1974, 2: 207—209
- 23 Haysae T, Humphery J A C, Grief A R. A consistently formulated QUICK scheme for fast and stable convergence using finite volume iterative calculation procedures. *J Comput Phys*, 1992, 93: 108—118[DOI]
- 24 Gray D D, Giorgini A. The validity of the Boussinesq approximation for liquids and gases. *Int J Heat Mass Transfer*, 1976, 19: 545—551
- 25 Ku H C, Hirsch R S, Taylor T D. A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations. *J Comput Phys*, 1987, 70: 439—462[DOI]
- 26 Guj G, Stella F. A vorticity-velocity method for the numerical solution of 3D incompressible flows. *J Comput Phys*, 1993, 106: 286—298
- 27 Tang L Q, Cheng T, Tsang T T H. Transient solutions for three-dimensional lid-driven cavity flows by a least-squares finite element method. *Int J Numer Methods Fluids*, 1995, 21: 413—432[DOI]
- 28 Fusegi T, Hyun J M, Kuwahara K, et al. A numerical study of three-dimensional natural convection in a differentially heated cubical enclosure. *Int J Heat Mass Transfer*, 1991, 34: 1543—1555[DOI]
- 29 Wakashima S, Saitoh T S. Benchmark solutions for natural convection in a cubic cavity using the high-order time-space method. *Int J Heat Mass Transfer*, 2004, 47: 853—864[DOI]