第 30 卷第 8 期 工 程 热 物 理 学 报 Vol.30, No.8
2009 年 8 月 JOURNAL OF ENGINEERING THERMOPHYSICS Aug., 2009

求解流动与传热问题的一种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，陕西 西安 710049)

摘 要 本文提出了一种求解流动与传热问题的高效稳定的分离式算法 —IDEAL(Inner Doubly-iterative Efficient Algorithm for Linked-equations). 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. 本文通过 2 个三维不可压缩流动和传热的算例对 IDEAL 算法与其它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. 在这 2 个算例中 IDEAL 算法几乎可以在任意的松弛因子下获得收敛的解，并且 IDEAL 算法所需最短计算时间较 SIMPLER 算法减少 12.9%~52.6%；较 SIMPLEC 算法减少 48.3%~79.1%；较 PISO 算法减少 10.7%~46.5%.

关键词 分离式算法；IDEAL；收敛速度；健壮性

中图分类号: TK124 **文献标识码**: A **文章编号**: 0253-231X(2009)08-1369-04

AN EFFICIENT SEGREGATED ALGORITHM FOR INCOMPRESSIBLE FLUID FLOW AND HEAT TRANSFER PROBLEMS—IDEAL

SUN Dong-Liang QU Zhi-Guo HE Ya-Ling TAO Wen-Quan

(*The State Key Lab. of Multiphase Flow in Power Engineering, Xi'an Jiaotong University, Xi'an 710049, China*)

Abstract Recently an efficient segregated solution procedure for incompressible fluid flow and heat transfer problems, called IDEAL (Inner Doubly-iterative Efficient Algorithm for Linked-equations), has been proposed by author. In the algorithm there exist inner doubly-iterative processes for pressure equation at each iteration level, which almost completely overcome two approximations in SIMPLE algorithm. Thus the coupling between velocity and pressure is fully guaranteed, greatly enhancing the convergence rate and stability of solution process. In the present paper the property of the IDEAL algorithm for three-dimensional incompressible fluid flow and heat transfer problems is analyzed by the comparison between the algorithm and three most widely-used algorithms (SIMPLER, SIMPLEC and PISO). By the comparison for two application examples we can find that the IDEAL algorithm, which can converge almost at any under-relaxation factor, is far more robust than SIMPLER, SIMPLEC and PISO algorithms. When each method uses its own optimal under-relaxation factor, the IDEAL algorithm can reduce the computation time by 12.9%~52.6% over SIMPLER algorithm, by 48.3%~79.1% over SIMPLEC algorithm and by 10.7%~46.5% over PISO algorithm.

Key words segregated algorithm; IDEAL; convergence rate; robustness

0 前 言

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. 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，所以该方法的得到了普遍的应用. 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[1] 提出，并被命名为 SIMPLE 算法. SIMPLE 算法有两个基本假设，(1) 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；(2) 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，但会影响计算

收稿日期: 2008-12-18; **修订日期**: 2009-06-15

基金项目: 国家自然科学基金重点项目 (No.50636050); 国家重点基础研究发展计划 (No.2006CB601203; No.2007CB206902)

作者简介: 孙东亮 (1977-)，男，河北张家口人，博士，主要从事强化传热和高效数值模拟方面的研究.

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

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，如 SIMPLER 算法 [2]，SIMPLEC 算法 [3]，PISO 算法 [4] 等. 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，为了获得压力修正方程，在速度修正方程中不可避免地忽略邻点速度修正的影响，因此为了克服 SIMPLE 算法中的第二个假设，我们需要直接改进压力，而不应引进压力修正值，以上思想是 CLEAR 算法 [5] 的成功所在. 但是直接求解压力方程会导致 CLEAR 算法稳定性较差，为了克服这一缺点本文在 CLEAR 算法的基础上提出了 IDEAL 算法 (Inner Doubly-Iterative Efficient Algorithm for Linked-Equations).

1 IDEAL 算法的提出

SIMPLE 算法的两个基本假设是影响计算收敛速度和计算过程稳定性的根本原因，所以一个理想的算法应同时克服以上两个基本假设，也就是同时具有以下两个基本特征：(1) 初始压力应尽可能接近当前迭代层次上的最终压力，该条件可通过克服 SIMPLE 算法的第一个假设获得. 这样在该初始压力下计算动量方程所得的中间速度基本同时满足了连续性方程，从而保证了算法的收敛性和健壮性；(2) 每个迭代层次的最终速度应该尽可能同时满足动量和连续性方程，该条件可以通过克服 SIMPLE 算法的第二个假设获得. 这是另外一个提高算法收敛性和健壮性的条件. 通过以上分析我们对压力方程进行两次内迭代计算，从而使得 IDEAL 算法同时具有以上两个基本特征，保证了算法的收敛性和健壮性.

2 IDEAL 算法的计算步骤

IDEAL 算法在三维交错网格下的计算步骤如下：

(1) 假设一个初始速度场 u^0, v^0 和 w^0.

(2) 利用初始速度场计算动量方程 (1)~(3) 的系数 a 和源项 b. 系数 a 和源项 b 的具体形式依赖于采用的离散格式，具体推导过程和计算公式可参考文献 [6].

(3) 由 u^0, v^0 和 w^0 计算三个假拟速度 $\tilde{u}^0$, $\tilde{v}^0$, $\tilde{w}^0$，它们构成下列三个动量方程的右端第一项：

$$u_e^{\mathrm{Temp}} = (\sum a_{nb}u_{nb}^0 + b)/(a_e/\alpha_u) + d_e(p_P^{\mathrm{Temp}} - p_E^{\mathrm{Temp}}) = \tilde{u}_e^0 + d_e(p_P^{\mathrm{Temp}} - p_E^{\mathrm{Temp}}) \quad (1)$$

$$v_n^{\mathrm{Temp}} = (\sum a_{nb}v_{nb}^0 + b)/(a_n/\alpha_v) + d_n(p_P^{\mathrm{Temp}} - p_N^{\mathrm{Temp}}) = \tilde{v}_n^0 + d_n(p_P^{\mathrm{Temp}} - p_N^{\mathrm{Temp}}) \quad (2)$$

$$w_t^{\mathrm{Temp}} = (\sum a_{nb}w_{nb}^0 + b)/(a_t/\alpha_w) + d_t(p_P^{\mathrm{Temp}} - p_T^{\mathrm{Temp}}) = \tilde{w}_t^0 + d_t(p_P^{\mathrm{Temp}} - p_T^{\mathrm{Temp}}) \quad (3)$$

(4) 求解压力方程 (4)，计算出临时压力 p^{Temp}.

$$(a_P/\alpha_p)p_P^{\mathrm{Temp}} = \sum a_{nb}p_{nb}^{\mathrm{Temp}} + b$$

$$a_P = a_E + a_W + a_N + a_S + a_T + a_B$$

$$a_E = (\rho Ad)_e,\ a_W = (\rho Ad)_w,\ a_N = (\rho Ad)_n \quad (4)$$

$$a_S = (\rho Ad)_s,\ a_T = (\rho Ad)_t,\ a_B = (\rho Ad)_b$$

$$b = (\rho\tilde{u}^0A)_w - (\rho\tilde{u}^0A)_e + (\rho\tilde{v}^0A)_s - (\rho\tilde{v}^0A)_n + (\rho\tilde{w}^0A)_b - (\rho\tilde{w}^0A)_t + (1-\alpha_p)\frac{a_P}{\alpha_p}p_P^{P\mathrm{Temp}}$$

压力方程 (4) 是通过把动量方程 (1)~(3) 代入连续性方程 (5) 而获得：

$$(\rho u)_eA_e - (\rho u)_wA_w + (\rho v)_nA_n - (\rho v)_sA_s + (\rho w)_tA_t - (\rho v)_bA_b = 0 \quad (5)$$

值得指出，在式 (4) 中对压力进行了亚松弛处理，并且将亚松弛组织到迭代计算过程中，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，即亚松弛因子常常可以取为 1.

(5) 把临时压力 p^{Temp} 代入式 (1)~(3) 计算出临时速度 u^{Temp}, v^{Temp}, w^{Temp}.

(6) 步骤 (4) 和 (5) 计算出的临时速度和压力看作上一次内迭代步的临时速度和压力，表示为 $u^{P\mathrm{Temp}}$, $v^{P\mathrm{Temp}}$, $w^{P\mathrm{Temp}}$ 和 $p^{P\mathrm{Temp}}$，并返回步骤 (3) 开始下一次内迭代步，同时步骤 (3) 和步骤 (4) 内上标为 "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，更好地满足速度和压力的内在联系的和谐.

(7) 根据求解出的初始压力 p^* 求解动量方程 (6)~(8)，获得中间速度 u^*, v^*, w^*.

$$(a_e/\alpha_u)u_e^* = \sum a_{nb}u_{nb}^* + b + A_e(p_P^* - p_E^*) \quad (6)$$

$$(a_n/\alpha_v)v_n^* = \sum a_{nb}v_{nb}^* + b + A_n(p_P^* - p_N^*) \quad (7)$$

$$(a_t/\alpha_w)w_t^* = \sum a_{nb}w_{nb}^* + b + A_t(p_P^* - p_T^*) \quad (8)$$

(8) 由 u^*, v^* 和 w^* 计算三个假拟速度 $\tilde{u}^*$, $\tilde{v}^*$, $\tilde{w}^*$，它们构成下列动量方程右端的第一项：

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$u_e^{\mathrm{Temp}}=(\sum a_{nb}u_{nb}^*+b)/(a_e/\alpha_u)+$$
$$d_e(p_P^{\mathrm{Temp}}-p_E^{\mathrm{Temp}})=\tilde{u}_e^*+d_e(p_P^{\mathrm{Temp}}-p_E^{\mathrm{Temp}}) \quad (9)$$
$$v_n^{\mathrm{Temp}}=(\sum a_{nb}v_{nb}^*+b)/(a_n/\alpha_v)+$$
$$d_n(p_P^{\mathrm{Temp}}-p_N^{\mathrm{Temp}})=\tilde{v}_n^*+d_n(p_P^{\mathrm{Temp}}-p_N^{\mathrm{Temp}}) \quad (10)$$
$$w_t^{\mathrm{Temp}}=(\sum a_{nb}w_{nb}^*+b)/(a_t/\alpha_w)+$$
$$d_t(p_P^{\mathrm{Temp}}-p_T^{\mathrm{Temp}})=\tilde{w}_t^*+d_t(p_P^{\mathrm{Temp}}-p_T^{\mathrm{Temp}}) \quad (11)$$

(9) 求解压力方程 (12), 计算出临时压力 p^{Temp}.

$$\begin{aligned}
&a_Pp_P^{\mathrm{Temp}}=\sum a_{nb}p_{nb}^{\mathrm{Temp}}+b\\
&a_P=a_E+a_W+a_N+a_S+a_T+a_B\\
&a_E=(\rho Ad)_e,\ a_W=(\rho Ad)_w,\ a_N=(\rho Ad)_n\\
&a_S=(\rho Ad)_s,\ a_T=(\rho Ad)_t,\ a_B=(\rho Ad)_b\\
&b=(\rho\tilde{u}^*A)_w-(\rho\tilde{u}^*A)_e+(\rho\tilde{v}^*A)_s-(\rho\tilde{v}^*A)_n+\\
&\quad(\rho\tilde{w}^*A)_b-(\rho\tilde{w}^*A)_t
\end{aligned} \quad (12)$$

在第二次内迭代过程中压力不需要亚松弛处理. 压力方程 (12) 是通过把动量方程 (9)~(11) 代入连续性方程 (5) 而获得.

(10) 把临时压力 p^{Temp} 代入式 (9)~(11) 计算出临时速度 u^{Temp}, v^{Temp}, w^{Temp}.

(11) 步骤 (9) 和 (10) 计算出的临时速度和压力看作上一次内迭代步的临时速度和压力, 表示为 $u^{P\mathrm{Temp}}$, $v^{P\mathrm{Temp}}$, $w^{P\mathrm{Temp}}$ 和 $p^{P\mathrm{Temp}}$, 并返回步骤 (8) 开始下一次内迭代步, 同时步骤 (8) 和步骤 (9) 内上标为 "*" 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 $N2$, 然后令第二次内迭代最终的速度作为当前迭代层次上求解所得的 u, v, w. 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能同时较好地满足连续性方程和隐式的动量方程.

(12) 求解其它标量方程.

(13) 把当前迭代层次上的最终速度 u, v, w 看作下一个迭代层次上的初始速度 u^0, v^0, w^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 算法的初始压力是通过求解压力方程获得的, 但是由于所获得的压力代数方程组是通过迭代求解的, 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. 通过计算实例发现在 IDEAL 算法中选用第一次内迭代所获得的压力作为下一个迭代层次上的初始压力场, 可以提高算法的收敛性和稳定性.

3 算例比较

以下我们在三维直角坐标情况对 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.

3.1 算例 1: 外掠后台阶流动

外掠后台阶流动属于开口系统, 三维的流动结构如图 1 所示. 计算在 Re =100~300, 网格数 =80×20×20~160 × 41×41 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}.

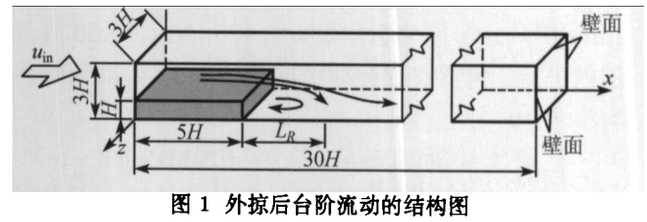

图 1 外掠后台阶流动的结构图

Fig. 1 Flow configuration over a backward-facing step

在图 2 中对 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 1 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.

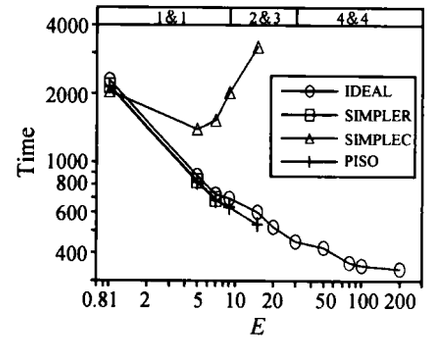

图 2 IDEAL 算法与 SIMPLER、SIMPLEC 和 PISO 算法在计算时间和健壮性方面的比较 (网格数 =160×41×41, Re=100)

Fig. 2 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms (grid number=160×41×41, Re=100)

如表 1 所示, IDEAL 算法较 SIMPLER 算法减少的计算时间为 49.6%~52.6%; IDEAL 算法较 SIMPLEC 算法减少的计算时间为 69.6%~79.1%; IDEAL 算法较 PISO 算法减少的计算时间为 36.3%~46.5%.

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

表 1 IDEAL 算法较 SIMPLER、SIMPLEC 和 PISO 算法最短计算时间的减少比率 (算例 1)

Table 1 Reducing ratio of computation time of IDEAL algorithm over SIMPLER, SIMPLEC and PISO algorithms

网格数	80×20×20		160×41×41
Re	100	300	100
相对 SIMPLER 减小比率	52.6%	49.6%	50.8%
相对 SIMPLEC 减小比率	79.1%	69.6%	75.8%
相对 PISO 减小比率	43.1%	46.5%	36.3%

3.2 算例 2: 方腔自然对流

方腔自然对流是典型的速度和压力耦合问题，三维正方形腔体的流动结构如图 3 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 $Ra=10^4\sim10^5$, 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}.

在图 4 中对 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它三种算法.

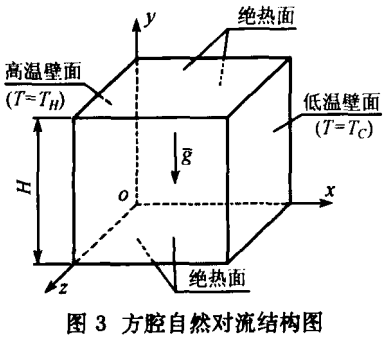

图 3 方腔自然对流结构图

Fig. 3 Flow configuration of natural convection in a cubic cavity

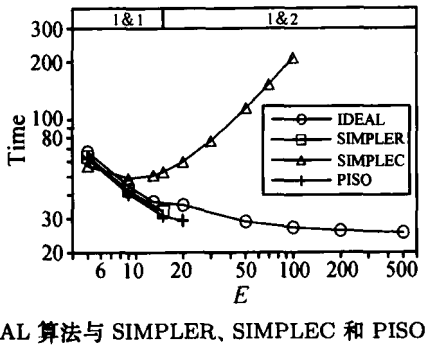

图 4 IDEAL 算法与 SIMPLER、SIMPLEC 和 PISO 算法在计算时间和健壮性方面的比较 (网格数 =30×30×30, $Ra=10^4$)

Fig. 4 Comparison of computation time and robustness of IDEAL, SIMPLER, SIMPLEC and PISO algorithms (grid number=30×30×30, $Ra=10^4$)

如表 2 所示，IDEAL 算法较 SIMPLER、SIMPLEC 和 PISO 算法减少的计算时间分别为 12.9%~22.4%、48.3%~63.7%和 10.7%~13.7%.

表 2 IDEAL 算法较 SIMPLER、SIMPLEC 和 PISO 算法最短计算时间的减少比率 (算例 2)

Table 2 Reducing ratio of computation time of IDEAL algorithm over SIMPLER, SIMPLEC and PISO algorithms

网格数	30×30×30	50×50×50
Ra	10^4	10^5
相对 SIMPLER 减小比率	22.4%	12.9%
相对 SIMPLEC 减小比率	48.3%	63.7%
相对 PISO 减小比率	13.7%	10.7%

4 结 论

本文提出了一种求解流动与传热问题的高效稳定的 IDEAL 算法. 本文对 IDEAL 算法与最广泛使用的三种算法 —SIMPLER、SIMPLEC 和 PISO 进行了三维问题的系统的比较分析，得出以下结论：

(1) IDEAL 算法在健壮性和收敛性方面皆优于其它三种算法.

(2) 在我们研究的 2 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.

(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分别为：IDEAL 算法较 SIMPLER 算法减少比率为 12.9%~52.6%; IDEAL 算法较 SIMPLEC 算法减少比率为 48.3%~79.1%; IDEAL 算法较 PISO 算法减少比率为 10.7%~46.5%.

参 考 文 献

[1] 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

[2] Patankar S V. A Calculation Procedure for Two-Dimensional Elliptic Situations. Numer. Heat Transfer, 1981, 4: 409-425

[3] Van Doormaal J P, Raithby G D. Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows. Numer. Heat Transfer, 1984, 7: 147-163

[4] Issa R I. Solution of Implicitly Discretized Fluid Flow Equation by Operator-Splitting. J. Comput. Phys., 1985, 62: 40-65

[5] 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

[6] Tao W Q. Numerical Heat Transfer. 2nd. Xi'an: Xi'an Jiaotong University Press, 2001

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net