

文章编号:1006-2467(2019)06-0681-07

DOI: 10.16183/j.cnki.jsjtu.2019.06.007

# 一种不可压缩二维流动的显式逐次超松弛并行算法

张晓慧, 柏君励, 顾解冲, 马宁

(上海交通大学 海洋工程国家重点实验室; 高新船舶与深海开发装备协同创新中心;  
船舶海洋与建筑工程学院, 上海 200240)

**摘要:** 提出一种有限体积显式逐次超松弛并行(FV-pSOR)算法,以提高逐次超松弛(SOR)算法求解不可压缩二维流动控制方程组离散所形成的代数方程组的效率.基于区域分解的思想,将计算域分割成4个子域,构造了离散的一般性代数方程组的显式迭代公式并规划了迭代路径;然后,通过数值求解典型二维方腔流,验证了FV-pSOR算法的有效性.结果表明:与SOR算法相比,所提FV-pSOR算法在计算精度相当的前提下的计算效率提高了数倍.

**关键词:** 分组显式;逐次超松弛;并行计算;有限体积;区域分解

**中图分类号:** U 661 **文献标志码:** A

## An Explicit Parallel Successive Over-Relaxation Method for Simulation of 2-Dimensional Incompressible Flows

ZHANG Xiaohui, BAI Junli, GU Xiechong, MA Ning

(State Key Laboratory of Ocean Engineering; Collaborative Innovation Center for Advanced Ship and Deep-Sea Exploration; School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

**Abstract:** An explicit finite volume parallel successive over-relaxation (FV-pSOR) iterative method is proposed to improve the efficiency of successive over-relaxation (SOR) iterative method in solving the algebraic equations obtained by discretizing the governing equations of 2-dimensional incompressible flows. Based on the domain decomposition technique, computation domain is separated into four sub-domains, and group explicit SOR schemes are constructed for the algebraic equations with general coefficients in flow problems. Furthermore, the iterative route has been discussed in detail. To confirm availability of this algorithm, a series of calculations for a typical lid-driven cavity flow are performed. It demonstrates that comparing to the SOR iterative method, current FV-pSOR iterative method improves the computation efficiency several folds while the accuracy is not degraded.

**Key words:** group explicit; successive over-relaxation (SOR); parallel computation; finite volume; domain decomposition

计算流体力学方法已逐渐成为流动及流体传热问题研究的主流工具.研究表明<sup>[1-4]</sup>,与势流方法相

比,采用计算流体力学方法能够更准确地预报船舶的耐波性.其中,主要方法是采用有限体积法

收稿日期:2017-10-24

基金项目:国家自然科学基金面上项目(51279105,BC0100097)

作者简介:张晓慧(1992-),女,陕西省渭南市人,博士生,研究方向为船舶耐波性的数值模拟.

通信作者:顾解冲,男,副教授,博士生导师,电话(Tel.)021-34206280; E-mail:xcgu@sjtu.edu.cn.

(FVM) 离散 Navier-Stokes 方程而形成代数方程组, 再进行迭代求解. 然而, 采用计算流体力学方法求解一个算例需要耗费大量时间, 尤其是迭代计算. 因为每个时间层涉及多次外迭代, 每次外迭代又涉及速度、压力或压力修正值的内迭代, 所以提高迭代计算的效率已经成为计算流体力学的一个关键技术问题, 而实行并行计算是提高计算效率的一条行之有效的途径. 随着分组显式思想和区域分解方法<sup>[5]</sup>的提出, 为并行计算创造了条件. 目前, 并行求解线性代数方程组的迭代方法主要包括 Schwarz 法、多重网格法、共轭梯度(CG)法以及经典的定常迭代法等. Schwarz 法基于区域分解的思想, 易于并行化但计算的收敛速度较慢. 例如, 文献[6]中通过乘性 Schwarz 方法以及预处理过程来加速迭代的收敛速度, 但其很难程序化. 多重网格方法<sup>[7]</sup>是通过将计算过程转换到粗网格中以加快迭代收敛速度. 另外, Krylov 子空间迭代法<sup>[8]</sup>包含了共轭梯度法及其扩展方法, 是另一类较为重要的迭代方法. 考虑到计算的效率和可操作性, 本文选取经典的定常迭代法. 虽然 Jacobi 迭代法容易实行并行操作, 但其比 Gauss-Seidel 和逐次超松弛(SOR)算法的收敛速度慢. 而相比于 Gauss-Seidel 算法, SOR 算法的收敛速度更快. 文献[9]中基于区域分解思想并结合 SOR 方法对 Stokes 方程进行并行计算.

最近, 许秋燕等<sup>[10]</sup>结合区域分解和分组显式方法将计算域分割成多个子域, 构造了子域并行迭代计算格式, 以用于有限差分求解特定的泊松方程, 最终求解的是常系数代数方程组. 而对于流动以及流体传热问题, 代数方程组的系数将随着外迭代和时间层的不断推进而不断变化.

本文采用船舶水动力学研究中常用的 FVM 求解不可压缩 Navier-Stokes 方程, 从求解二维流体流动入手, 在许秋燕等<sup>[10]</sup>研究的基础上, 提出一种基于 FVM 的显式逐次超松弛并行算法(FV-pSOR). 其中, 采用 4 阶紧致格式离散流项和扩散项、4 阶龙格-库塔法步进非定常项以及同位网格 SIMPLE (Semi-Implicit Method for Pressure-Linked Equation) 算法求解压力-速度耦合方程<sup>[11]</sup>. 基于区域分解, 将计算域分割成 4 个子域而构造所需分组显式 SOR 迭代公式, 并通过 3 种网格密度下典型二维驱动方腔流的计算来验证所提 FV-pSOR 算法的有效性.

## 1 数学模型

### 1.1 Navier-Stokes 方程组及其数值离散

不可压缩流体的流动服从如下控制方程组:

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} \quad (2)$$

式中:  $\mathbf{u}$  为速度矢量;  $\nabla$  为哈密顿梯度算子;  $t$  为时间;  $\rho$  为流体密度;  $p$  为压力;  $\nu$  为流体运动学黏度.

本文并不是直接求解微分形式的控制方程组(式(1)、(2)), 而是求解在同位网格中由 FVM 导出的积分形式的控制方程<sup>[11]</sup>. 同位网格是将速度、压力、黏度等物理量均定义在网格中心点处, 如图 1 所示. 其中, 网格 P 的积分方程为

$$\frac{\partial \mathbf{u}_{m+1}}{\partial t} + \mathbf{F}_c = -\frac{1}{\rho} \nabla p_{m+1} + \mathbf{F}_D \quad (3)$$

式中:  $\mathbf{F}_c$  为对流项;  $\mathbf{F}_D$  为扩散项.

对流项可用前一时刻的速度进行线性化, 即

$$\mathbf{F}_c = \frac{C^e \mathbf{u}_{m+1}^e - C^w \mathbf{u}_{m+1}^w}{\Delta x} + \frac{C^t \mathbf{u}_{m+1}^t - C^b \mathbf{u}_{m+1}^b}{\Delta z} \quad (4)$$

式中:  $C$  为前一时刻的速度,  $C^e = u_m^e$ ,  $C^w = u_m^w$ ,  $C^t = u_m^t$ ,  $C^b = u_m^b$ ;  $\Delta x$  和  $\Delta z$  表示沿水平和垂直方向的网格尺寸;  $m$  为前一时间步;  $m+1$  为当前时间步;  $E, e, W, w, B, b, T, t$  均表示控制体的界面(见图 1).

扩散项为

$$\mathbf{F}_D = \nu \left[ \left( \frac{\partial \mathbf{u}}{\partial x} \right)_e - \frac{\partial \mathbf{u}}{\partial x} \right]_w / \Delta x + \left( \frac{\partial \mathbf{u}}{\partial z} \right)_t - \frac{\partial \mathbf{u}}{\partial z} \Big|_b / \Delta z \quad (5)$$

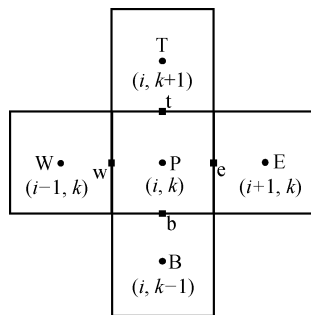


图 1 同位网格

Fig. 1 Collocated grid

式(4)、(5)中的变量包括速度和速度导数, 采用 4 阶紧致格式插值<sup>[11]</sup>. 式(3)中时间层的步进采用 4 阶龙格-库塔法<sup>[11]</sup>, 龙格-库塔法的每一步都采用 SIMPLE 算法求解速度-压力耦合方程.

经上述离散后, 最终所得压力修正方程为

$$a_{Pi,k} \phi_{i,k} = b_{i,k} + a_{i,k}^P \phi_{i+1,k} + a_{i,k}^W \phi_{i-1,k} + a_{i,k}^T \phi_{i,k+1} + a_{i,k}^B \phi_{i,k-1} \quad (6)$$

式中:  $\phi$  为压力修正值, 并略去代表时间步的上标;  $a_{Pi,k}, b_{i,k}, a_{i,k}^P, a_{i,k}^W, a_{i,k}^T, a_{i,k}^B$  均为系数, 其具体表达式详见文献[11].

下面将具体说明式(6)的显式 SOR 并行迭代算

法的构造过程.

## 1.2 FV-pSOR 算法的基础迭代格式

FV-pSOR 算法有 4 种基础 SOR 迭代格式,如图 2 所示.图中,横向代表第 1 个下标  $i$ ,垂向代表第 2 个下标  $k$ ,箭头表示迭代计算的行进方向.对于式(6),4 种基础 SOR 迭代格式的具体表达式如下:

基础迭代格式 A1 为

$$\begin{aligned} \phi_{i,k}^{n+1} &= (1-\omega)\phi_{i,k}^n + \frac{\omega}{a_{i,k}^P}(b_{i,k} + a_{i,k}^E\phi_{i+1,k}^n + \\ &a_{i,k}^W\phi_{i-1,k}^{n+1} + a_{i,k}^T\phi_{i,k+1}^n + a_{i,k}^B\phi_{i,k-1}^{n+1}) \end{aligned} \quad (7)$$

基础迭代格式 A2 为

$$\begin{aligned} \phi_{i,k}^{n+1} &= (1-\omega)\phi_{i,k}^n + \frac{\omega}{a_{i,k}^P}(b_{i,k} + a_{i,k}^E\phi_{i+1,k}^n + \\ &a_{i,k}^W\phi_{i-1,k}^{n+1} + a_{i,k}^T\phi_{i,k+1}^{n+1} + a_{i,k}^B\phi_{i,k-1}^n) \end{aligned} \quad (8)$$

基础迭代格式 A3 为

$$\begin{aligned} \phi_{i,k}^{n+1} &= (1-\omega)\phi_{i,k}^n + \frac{\omega}{a_{i,k}^P}(b_{i,k} + a_{i,k}^E\phi_{i+1,k}^{n+1} + \\ &a_{i,k}^W\phi_{i-1,k}^n + a_{i,k}^T\phi_{i,k+1}^n + a_{i,k}^B\phi_{i,k-1}^{n+1}) \end{aligned} \quad (9)$$

基础迭代格式 A4 为

$$\begin{aligned} \phi_{i,k}^{n+1} &= (1-\omega)\phi_{i,k}^n + \frac{\omega}{a_{i,k}^P}(b_{i,k} + a_{i,k}^E\phi_{i+1,k}^{n+1} + \\ &a_{i,k}^W\phi_{i-1,k}^n + a_{i,k}^T\phi_{i,k+1}^{n+1} + a_{i,k}^B\phi_{i,k-1}^n) \end{aligned} \quad (10)$$

式中:  $\omega$  为超松弛因子,其最佳取值为  $\omega = 2/(1 + \sqrt{1-r^2})$ ,  $r$  为对应的 Jacobi 矩阵的谱半径. 本文根据经验取  $\omega = 1.90$  进行迭代计算.

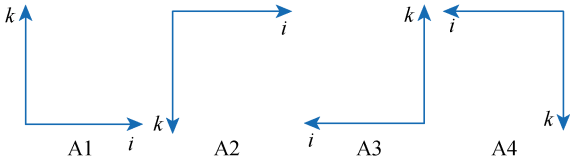


图 2 4 种基础迭代格式

Fig. 2 Four iterative schemes

根据分组显式的思想,由上述 4 种基础 SOR 迭代格式构造的迭代格式 A5 见图 3. 在节点  $(i,k+1)$  处采用基础格式 A3, 节点  $(i,k)$  处采用基础格式 A4, 节点  $(i+1,k)$  处采用基础格式 A2, 节点  $(i+1,k+1)$  处采用基础格式 A1. 因为当前第  $n$  次迭代的值已知,所以只有 4 个节点处第  $n+1$  次迭代的值为

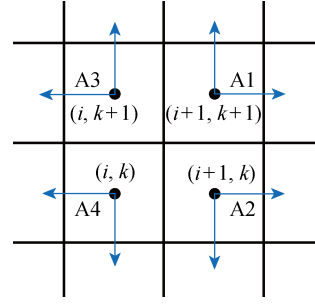


图 3 基本迭代格式 A5 示意图

Fig. 3 Sketch of scheme A5

未知,可由以下 4 个基础格式构造,即

$$\left. \begin{aligned} \phi_{i,k}^{n+1} &= a_1 + b_1\phi_{i+1,k}^{n+1} + b_2\phi_{i,k+1}^{n+1} \\ \phi_{i,k+1}^{n+1} &= a_2 + b_3\phi_{i+1,k+1}^{n+1} + b_4\phi_{i,k}^{n+1} \\ \phi_{i+1,k}^{n+1} &= a_3 + b_5\phi_{i,k}^{n+1} + b_6\phi_{i+1,k+1}^{n+1} \\ \phi_{i+1,k+1}^{n+1} &= a_4 + b_7\phi_{i,k+1}^{n+1} + b_8\phi_{i+1,k}^{n+1} \end{aligned} \right\} \quad (11)$$

联立求解可得

$$a_1 = (1-\omega)\phi_{i,k}^n + \frac{\omega}{a_{i,k}^P}(b_{i,k} + a_{i,k}^W\phi_{i-1,k}^n + a_{i,k}^B\phi_{i,k-1}^n)$$

$$a_2 = (1-\omega)\phi_{i,k+1}^n + \frac{\omega}{a_{i,k+1}^P}(b_{i,k+1} + a_{i,k+1}^W\phi_{i-1,k+1}^n + a_{i,k+1}^T\phi_{i,k+2}^n)$$

$$a_3 = (1-\omega)\phi_{i+1,k}^n + \frac{\omega}{a_{i+1,k}^P}(b_{i+1,k} + a_{i+1,k}^E\phi_{i+2,k}^n + a_{i+1,k}^B\phi_{i+1,k-1}^n)$$

$$a_4 = (1-\omega)\phi_{i+1,k+1}^n + \frac{\omega}{a_{i+1,k+1}^P}(b_{i+1,k+1} + a_{i+1,k+1}^E\phi_{i+2,k+1}^n + a_{i+1,k+1}^T\phi_{i+1,k+2}^n)$$

$$b_1 = \frac{\omega}{a_{i,k}^P}a_{i,k}^E, \quad b_2 = \frac{\omega}{a_{i,k}^P}a_{i,k}^T$$

$$b_3 = \frac{\omega}{a_{i,k+1}^P}a_{i,k+1}^E, \quad b_4 = \frac{\omega}{a_{i,k+1}^P}a_{i,k+1}^B$$

$$b_5 = \frac{\omega}{a_{i+1,k}^P}a_{i+1,k}^W, \quad b_6 = \frac{\omega}{a_{i+1,k}^P}a_{i+1,k}^T$$

$$b_7 = \frac{\omega}{a_{i+1,k+1}^P}a_{i+1,k+1}^W, \quad b_8 = \frac{\omega}{a_{i+1,k+1}^P}a_{i+1,k+1}^B$$

从而,由式(11)所得迭代格式 A5 的显式表达式为

$$\begin{bmatrix} \phi_{i,k}^{n+1} & \phi_{i,k+1}^{n+1} & \phi_{i+1,k}^{n+1} & \phi_{i+1,k+1}^{n+1} \end{bmatrix}^T = \mathbf{M} \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix}^T \quad (12)$$

式中:

$$\mathbf{M} = \begin{bmatrix} -\frac{1-b_3b_7-b_6b_8}{k} & -\frac{b_2+b_1b_6b_7-b_2b_6b_8}{k} & -\frac{b_1-b_1b_3b_7+b_2b_3b_8}{k} & -\frac{b_2b_3+b_1b_6}{k} \\ -\frac{b_4+b_3b_5b_8-b_4b_6b_8}{k} & -\frac{1-b_1b_5-b_6b_8}{k} & -\frac{b_1b_4+b_3b_8}{k} & -\frac{b_3-b_1b_3b_5+b_1b_4b_6}{k} \\ -\frac{b_5-b_3b_5b_7+b_4b_6b_7}{k} & -\frac{b_2b_5+b_6b_7}{k} & -\frac{1-b_2b_4-b_3b_7}{k} & -\frac{b_2b_3b_5+b_6-b_2b_4b_6}{k} \\ -\frac{b_4b_7+b_5b_8}{k} & -\frac{b_7-b_1b_5b_7+b_2b_5b_8}{k} & -\frac{b_1b_4b_7+b_8-b_2b_4b_8}{k} & -\frac{1-b_2b_4-b_1b_5}{k} \end{bmatrix}$$

$$k = -1 + b_3 b_7 + b_1 (b_5 - b_3 b_5 b_7 + b_4 b_6 b_7) + b_6 b_8 + b_2 (b_4 + b_3 b_5 b_8 - b_4 b_6 b_8)$$

同样地,由基础 SOR 格式可以构造图 4 所示的另外 4 种基础迭代格式. 鉴于这 4 种基础迭代格式的构造方法类似,本文仅说明格式 A6 的构造过程. 在节点  $(i, k)$  处采用基础格式 A3, 节点  $(i+1, k)$  处采用基础格式 A1, 对应的迭代方程组为

$$\left. \begin{aligned} \phi_{i,k}^{n+1} &= c_1 + b_1 \phi_{i+1,k}^{n+1} \\ \phi_{i+1,k}^{n+1} &= c_2 + b_5 \phi_{i,k}^{n+1} \end{aligned} \right\} \quad (13)$$

式中:

$$\begin{aligned} c_1 &= (1-\omega) \phi_{i,k}^n + \frac{\omega}{a_{i,k}^p} (b_{i,k} + a_{i,k}^w \phi_{i-1,k}^n + a_{i,k}^T \phi_{i,k+1}^n + a_{i,k}^B \phi_{i,k-1}^n) \\ c_2 &= (1-\omega) \phi_{i+1,k}^n + \frac{\omega}{a_{i+1,k}^p} (b_{i+1,k} + a_{i+1,k}^E \phi_{i+2,k}^n + a_{i+1,k}^T \phi_{i+1,k+1}^n + a_{i+1,k}^B \phi_{i+1,k-1}^n) \end{aligned}$$

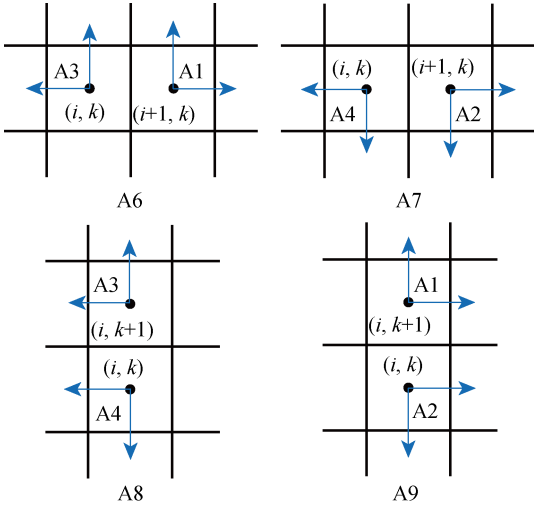


图 4 基本迭代格式 A6、A7、A8 及 A9 示意图  
Fig. 4 Sketch of scheme A6, A7, A8 and A9

求解式(13), 所得格式 A6 的显式表达式为

$$\left. \begin{aligned} \phi_{i,k}^{n+1} &= \frac{1}{1-b_1 b_5} (c_1 + b_1 c_2) \\ \phi_{i+1,k}^{n+1} &= \frac{1}{1-b_1 b_5} (c_2 + b_5 c_1) \end{aligned} \right\} \quad (14)$$

同样地,可以得到其他 3 种迭代格式. 其中, 格式 A7 为

$$\left. \begin{aligned} \phi_{i,k}^{n+1} &= \frac{1}{1-b_1 b_5} (c_3 + b_1 c_4) \\ \phi_{i+1,k}^{n+1} &= \frac{1}{1-b_1 b_5} (c_4 + b_5 c_3) \end{aligned} \right\} \quad (15)$$

式中:

$$\begin{aligned} c_3 &= (1-\omega) \phi_{i,k}^n + \frac{\omega}{a_{i,k}^p} (b_{i,k} + a_{i,k}^w \phi_{i-1,k}^n + a_{i,k}^T \phi_{i,k+1}^{n+1} + a_{i,k}^B \phi_{i,k-1}^n) \\ c_4 &= (1-\omega) \phi_{i+1,k}^n + \frac{\omega}{a_{i+1,k}^p} (b_{i+1,k} + a_{i+1,k}^E \phi_{i+2,k}^n + a_{i+1,k}^T \phi_{i+1,k+1}^{n+1} + a_{i+1,k}^B \phi_{i+1,k-1}^n) \end{aligned}$$

$$\begin{aligned} c_4 &= (1-\omega) \phi_{i+1,k}^{n+1} + \frac{\omega}{a_{i+1,k}^p} (b_{i+1,k} + a_{i+1,k}^E \phi_{i+2,k}^n + a_{i+1,k}^T \phi_{i+1,k+1}^{n+1} + a_{i+1,k}^B \phi_{i+1,k-1}^n) \end{aligned}$$

格式 A8 为

$$\left. \begin{aligned} \phi_{i,k}^{n+1} &= \frac{1}{1-b_2 b_4} (c_5 + b_2 c_6) \\ \phi_{i,k+1}^{n+1} &= \frac{1}{1-b_2 b_4} (c_6 + b_4 c_5) \end{aligned} \right\} \quad (16)$$

式中:

$$\begin{aligned} c_5 &= (1-\omega) \phi_{i,k}^n + \frac{\omega}{a_{i,k}^p} (b_{i,k} + a_{i,k}^E \phi_{i+1,k}^{n+1} + a_{i,k}^w \phi_{i-1,k}^n + a_{i,k}^B \phi_{i,k-1}^n) \\ c_6 &= (1-\omega) \phi_{i,k+1}^n + \frac{\omega}{a_{i,k+1}^p} (b_{i,k+1} + a_{i,k+1}^E \phi_{i+1,k+1}^{n+1} + a_{i,k+1}^w \phi_{i-1,k+1}^n + a_{i,k+1}^T \phi_{i,k+2}^n) \end{aligned}$$

格式 A9 为

$$\left. \begin{aligned} \phi_{i,k}^{n+1} &= \frac{1}{1-b_2 b_4} (c_7 + b_2 c_8) \\ \phi_{i,k+1}^{n+1} &= \frac{1}{1-b_2 b_4} (c_8 + b_4 c_7) \end{aligned} \right\} \quad (17)$$

式中:

$$\begin{aligned} c_7 &= (1-\omega) \phi_{i,k}^n + \frac{\omega}{a_{i,k}^p} (b_{i,k} + a_{i,k}^E \phi_{i+1,k}^n + a_{i,k}^w \phi_{i-1,k}^{n+1} + a_{i,k}^B \phi_{i,k-1}^n) \\ c_8 &= (1-\omega) \phi_{i,k+1}^n + \frac{\omega}{a_{i,k+1}^p} (b_{i,k+1} + a_{i,k+1}^E \phi_{i+1,k+1}^n + a_{i,k+1}^w \phi_{i-1,k+1}^{n+1} + a_{i,k+1}^T \phi_{i,k+2}^n) \end{aligned}$$

至此,所需显式 SOR 迭代格式全部构造完成.

## 2 迭代计算过程

图 5 和 6 示出了偶次迭代和奇次迭代的行进顺序. 图中, 计算域被分割成 4 个子域. 由图 5 可见, 偶次迭代顺序是顺次从中心向外行进的. 其迭代步骤如下:

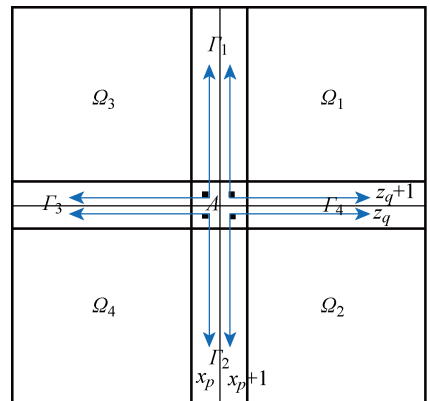


图 5 偶次迭代行进过程示意图

Fig. 5 Sketch of the process of even iteration

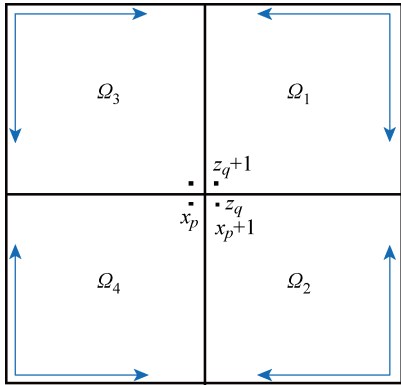


图 6 奇次迭代行进过程示意图  
Fig. 6 Sketch of the process of odd iteration

(1) 选定两个坐标方向的中点  $x_p, x_p + 1$  和  $z_q, z_q + 1$ , 共有 4 个位置点, 区域  $\Delta$  由 4 个选定的点构成. 按照格式 A5 的迭代公式算出这 4 个点在下一个迭代步的数值.

(2) 根据箭头的行进方向, 分别在区域  $\Gamma_1, \Gamma_2, \Gamma_3$  和  $\Gamma_4$  中按照格式 A6、A7、A8 和 A9 的迭代公式依次计算当前区域中的点在下一个迭代步的数值. 以  $\Gamma_1$  为例, 计算次序为

$$i = x_p, \quad x_p + 1$$
$$k = z_q + 2, z_q + 3, \dots, k_{\max}$$

其中:  $k_{\max}$  为  $k$  方向的最大网格数.

(3) 在区域  $\Omega_1, \Omega_2, \Omega_3$  和  $\Omega_4$  中分别按照格式 A1、A2、A3 和 A4 的迭代公式依次计算各区域中的点在下一个迭代步的数值.

完成上述 3 步迭代后, 所有点处的值均已得到更新.

如图 6 所示, 奇次迭代的迭代顺序是从四角向中心行进. 在区域  $\Omega_1, \Omega_2, \Omega_3$  和  $\Omega_4$  中分别按照格式 A4、A3、A2 和 A1 的迭代公式依次计算各区域中的点在下一个迭代步的数值. 迭代完成后, 所有点处的值均已得到更新.

至此, 偶次迭代和奇次迭代的过程构建完成. 计算时, 奇次迭代和偶次迭代交替进行, 直至精度满足控制要求, 然后, 进入下一个外迭代层或下一个时间层.

3 算例

以二维驱动方腔流为例来验证 FV-pSOR 算法的有效性. 图 7 所示为二维方腔流在左右侧面和底面需满足的固壁无滑移条件, 其顶面为给定的自左向右均匀剪切流动. 水平速度设为  $u = 1 \text{ m/s}$ , 垂向速度为  $v = 0$ .

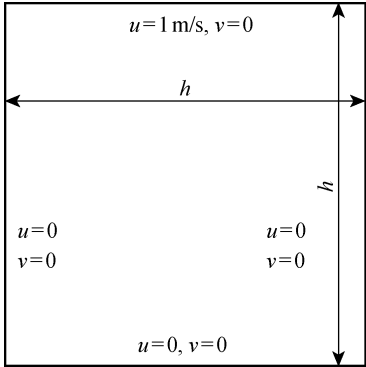


图 7 二维方腔流边界条件  
Fig. 7 Boundary conditions of 2D lid-driven cavity flow

取流动区域的边长为  $h = 1 \text{ m}$ , 流动的雷诺数  $Re = uh/\nu = 1\,000$ . 计算时, 将矩形计算域划分为均匀网格, 网格数量  $N$  分别为  $50 \times 50$ 、 $100 \times 100$  以及  $150 \times 150$ . 并行计算时, 将矩形计算域分割成 4 个子域. 本文分析方腔流发展到 35 s 左右时的计算结果. 采用一台处理器为 Intel 酷睿 i7-6700 的计算机, 调用 DATE\_AND\_TIME 函数所获两种算法在 3 种网格数量下的计算耗时见表 1. 可见: 在网格数量  $N = 50 \times 50$  下, SOR 算法的耗时超出 FV-pSOR 算法耗时 12 倍; 在网格分别为  $100 \times 100$  和  $150 \times 150$  下, SOR 算法的耗时分别超出 FV-pSOR 算法的耗时 6 和 7 倍. 可见, 对于相同的子域分割, 采用细网格有利于提高计算效率.

图 8~10 为网格数量分别为  $50 \times 50$ 、 $100 \times 100$  以及  $150 \times 150$  下的水平速度分布云图、垂向速度  $v$  沿  $y = 0.5 \text{ m}$  水平线的分布图、水平速度  $u$  沿  $x = 0.5 \text{ m}$  垂直线的分布图. 其中, 3 种网格数量下的云图分辨率相同. 为了便于对比, 图 9 和 10 中同时给

表 1 网格数量分别为 50×50、100×100 及 150×150 时的计算耗时  
Tab. 1 Time consumption under 50×50, 100×100 and 150×150 grids

算法	耗时/s		
	$N=50 \times 50$	$N=100 \times 100$	$N=150 \times 150$
FV-pSOR	1 515	9 796	43 626
SOR	19 635	62 193	343 072

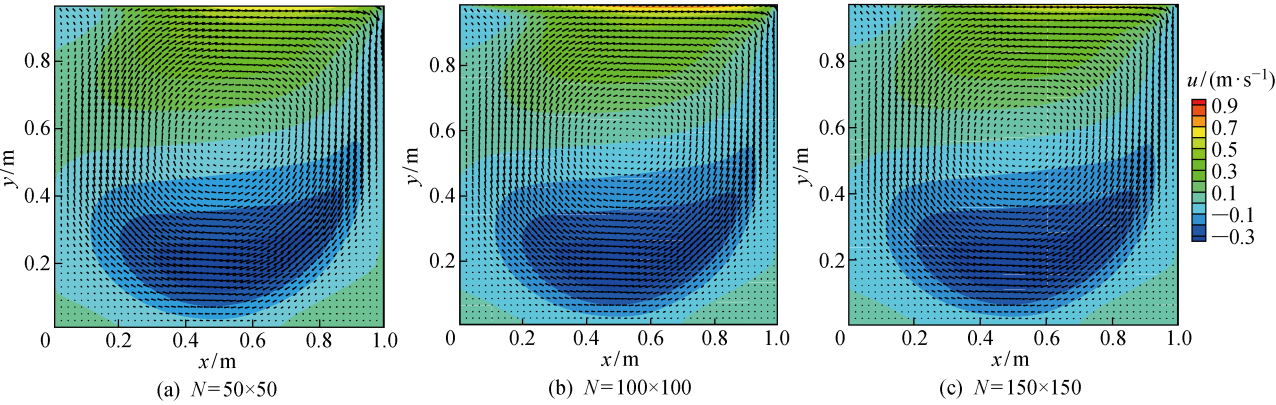


图 8 水平速度  $u$  的分布云图

Fig. 8 Distribution of velocity component  $u$

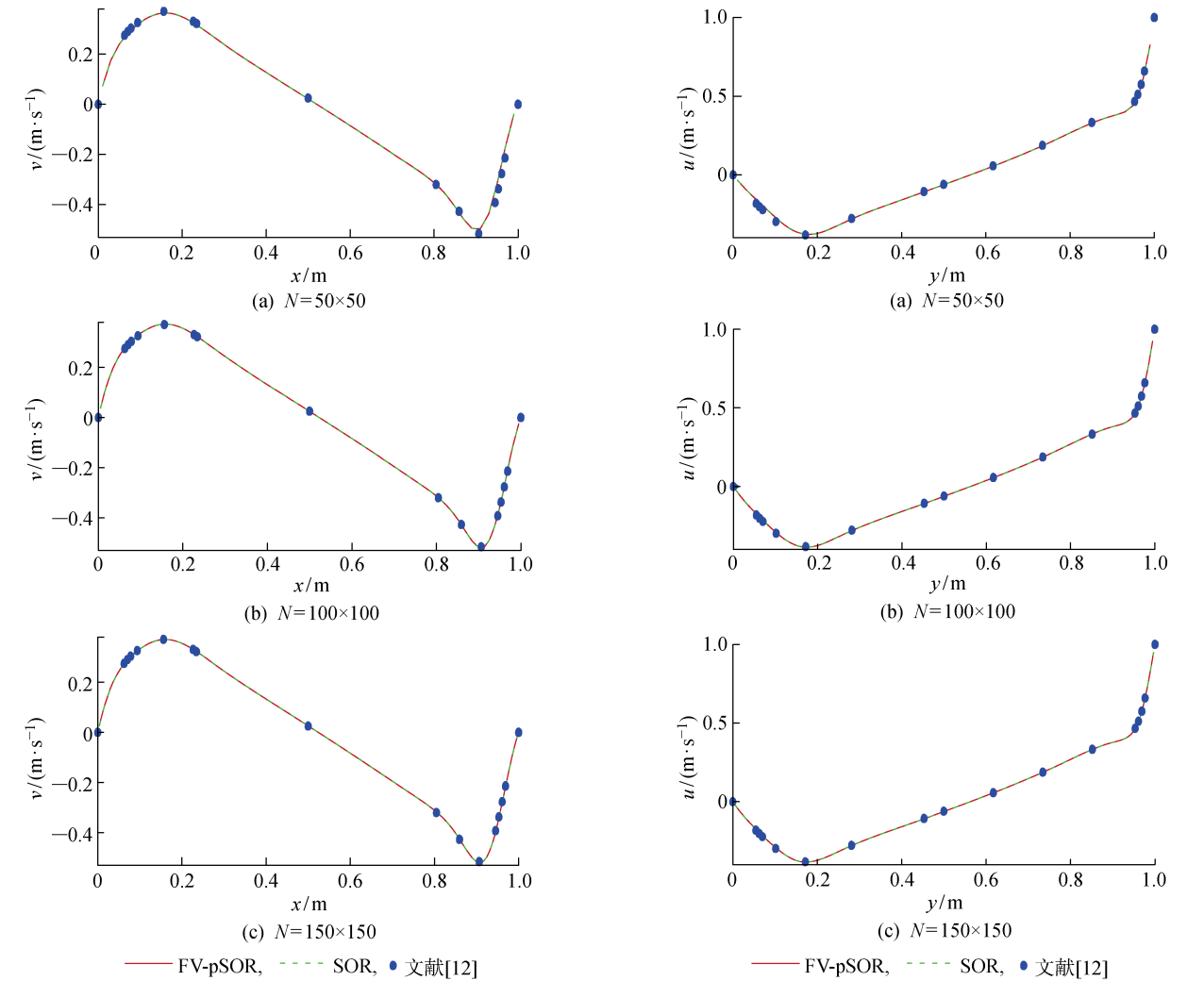


图 9 垂向速度  $v$  沿水平线  $y=0.5\text{ m}$  的分布情况  
Fig. 9 Velocity  $v$  along horizontal direction  $y=0.5\text{ m}$

图 10 水平速度  $u$  沿垂直线  $x=0.5\text{ m}$  的分布情况  
Fig. 10 Velocity  $u$  along vertical direction  $x=0.5\text{ m}$

出了 SOR 算法和文献[12]中的速度分布结果. 可以看出, 本文提出的 FV-pSOR 算法的结果与 SOR 算法的结果完全相同(差值为  $10^{-4}$  量级), 而且与文献[12]中的结果相符, 表明本文提出的 FV-pSOR 算

法与 SOR 算法具有同等计算精度. 由图 9 和 10 可以看出, 本文提出的 FV-pSOR 算法与 SOR 算法的计算精度差异不大. 而由表 1 可见, 其计算效率是 SOR 算法计算效率的 6~7 倍, 说明 FV-pSOR 算法

能够适应数值求解 Navier-Stokes 方程的五对角线性方程组的并行迭代求解.

综上所述,本文提出的 FV-pSOR 算的计算效率明显高于 SOR 算法,但在细网格条件下其计算效率的提高幅度比粗网格下的有所降低.

4 结语

本文在有限差分并行迭代算法的基础上提出了 FV-pSOR 算法,用于提高 SOR 方法求解 Navier-Stokes 方程的计算效率,并通过一个典型的二维驱动方腔流算例的计算,验证了 FV-pSOR 算法的有效性.结果表明,FV-pSOR 算法的计算效率较高,且其计算精度与 SOR 算法差异不大.虽然 FV-pSOR 算法的计算格式是基于计算流体力学流动问题的数值求解提出的,但其同样适用于其他五对角线性方程组的并行迭代求解.

参考文献:

[ 1 ] GUO Z, MA Q, HU X. Seakeeping analysis of a wave-piercing catamaran using URANS-based method [J]. **International Journal of Offshore & Polar Engineering**, 2016, 26(1): 48-56.

[ 2 ] TEZDOGAN T, INCECIK A, TURAN O. Full-scale unsteady RANS simulations of vertical ship motions in shallow water[J]. **Ocean Engineering**, 2016, 123: 131-145.

[ 3 ] ZHOU Y H, MA N, SHI X, *et al.* Direct calculation method of roll damping based on three-dimensional CFD approach[J]. **Journal of Hydrodynamics**, 2015, 27(2): 176-186.

[ 4 ] AVALOS G O G, WANDERLEY J B V, FERNANDES A C, *et al.* Roll damping decay of a FPSO with bilge keel[J]. **Ocean Engineering**, 2014, 87: 111-120.

[ 5 ] KEYES D E, GROPP W D. A comparison of domain

decomposition techniques for elliptic partial differential equations and their parallel implementation [J]. **SIAM Journal on Scientific and Statistical Computing**, 1987, 8(2): 166-202.

[ 6 ] WALKER E, NIKITOPOULOS D, TROMEUR-DEROUT D. Parallel solution methods for Poisson-like equations in two-phase flows[J]. **Computers & Fluids**, 2013, 80: 152-157.

[ 7 ] NOTAY Y, NAPOV A. A massively parallel solver for discrete Poisson-like problems [J]. **Journal of Computational Physics**, 2015, 281: 237-250.

[ 8 ] XU K L, YANG P, JIANG Y L. Structure-preserving model reduction of second-order systems by Krylov subspace methods[J]. **Journal of Applied Mathematics & Computing**, 2017(5): 1-18.

[ 9 ] LEMMER A, HILFER R. Parallel domain decomposition method with non-blocking communication for flow through porous media[J]. **Journal of Computational Physics**, 2015, 281: 970-981.

[10] 许秋燕, 张现强. 求解泊松方程的多子域超松弛并行迭代算法[J]. **宁夏大学学报(自然科学版)**, 2013, 34(2): 97-100.

XU Qiuyan, ZHANG Xianqiang. A multi-subdomain parallel super over relaxation iterative algorithm for solving Poisson equation [J]. **Journal of Ningxia University (Natural Science Edition)**, 2013, 34(2): 97-100.

[11] BAI J L, MA N, GU X C. Fourth order compact finite volume scheme for water wave simulation[C]// **The 18th Academic Symposium on China's Marine Engineering**. Zhoushan, China: Chinese Society of Oceanography, 2017: 101-106.

[12] GHIA U, GHIA K N, SHIN C T. High-*Re* solutions for incompressible flow using the Navier-Stokes equations and a multigrid method [J]. **Journal of Computational Physics**, 1982, 48(3): 387-411.

(本文编辑:何雪)