

基于多 GPU 的格子 Boltzmann 法 对槽道湍流的直接数值模拟*

许 丁, 陈 刚, 王 娴, 李跃明

(西安交通大学 航天航空学院, 机械结构强度与振动国家重点实验室, 西安 710049)

摘要: 采用多 GPU 并行的格子 Boltzmann 方法(lattice Boltzmann method, LBM)对充分发展的槽道湍流进行了直接数值模拟。GPU(graphic processing unit)的数据并行单指令多线程(single-instruction multiple-thread, SIMT)特征与 LBM 完美的并行性相匹配,使得 LBM 求解器在 GPU 上运行获得了极高的性能,亦使得大规模 DNS(direct numerical simulation)在桌面级计算机上进行成为可能。采用 8 个 GPU, 网格数目达到 6.7×10^7 , 全场网格尺寸 $\Delta^+ = 1.41$ 。模拟 3×10^6 个时间步长, 用时仅 24 h。另外, 直接模拟结果无论是在平均流速或湍流统计量上均与 Moser 等的结果吻合得很好, 这也证实了二阶精度的格子 Boltzmann 法直接模拟湍流的能力与有效性。

关键词: 格子 Boltzmann 法; 多 GPU 并行; 槽道湍流; 直接数值模拟(DNS)

中图分类号: TB126; O351 **文献标志码:** A

DOI: 10.3879/j.issn.1000-0887.2013.09.009

引 言

格子 Boltzmann 方法(LBM)是目前计算流体中最新的研究方法之一。它采用微观粒子模型,用离散的模型模拟连续的流体运动,不同于传统的 Navier-Stokes 方程求解方法,具有清晰的物理背景。该方法有 3 个其它模拟方法无法比拟的优点,即算法简单、完美的质量守恒性及天生的并行性^[1]。另外,该方法在边界处理上也相对简单,使得它处于传统方法无法胜任的领域,如微尺度流动与换热、多孔介质、生物流体、磁流体、燃烧、晶体生长等领域都可以进行有效模拟。可以说格子 Boltzmann 方法不仅仅是一种数值模拟方法,而且是一项重要科学研究手段^[2]。虽然,格子 Boltzmann 方法已被用于很多领域,但在模拟湍流的有效性方面,还有待进一步验证^[3-4]。为此,首先应该确认 LBM 在直接模拟湍流方面的能力。

充分发展的槽道湍流是一个看似简单,实则复杂的经典问题。长期以来,它以简单的几何吸引了大量学者对其内部湍流及边界层进行实验及理论分析,由此产生了大量的相关论文。其中以 Moser 等发表的 DNS 结果最为著名^[5-6],其 $Re_\tau = 180$ 的 DNS 结果一直以来作为数值模拟

* 收稿日期: 2013-05-30 修订日期: 2013-06-05

基金项目: 国家自然科学基金资助项目(11242010;11102150);中央高校基本科研业务费专项资金资助项目

作者简介: 许丁(1980—),男,西安市人,讲师,博士(E-mail:dingxu@mail.xjtu.edu.cn);
王娴(1977—),女,吉林省吉林市人,副教授,博士(通讯作者, E-mail:wangxian@mail.xjtu.edu.cn)。

的比较基准。随着计算机的发展,该问题相关的数值模拟工作越来越多。槽道湍流常被用来作为发展与检验湍流模型正确性的经典算例。然而,可被视为比较基准的结果却很少。由于受到计算机水平的限制,Moser 等^[5]的 DNS 网格数仅为 128^3 ,非均分最小网格尺寸 $\Delta^+ = 4.4$,远大于当地 Kolmogorov 长度。且其所用的 KMM 程序是专门针对该问题开发的,对于常用的数值求解方法,该网格数目是远远不够的。即使在计算机高速发展的今天,该问题的 DNS 对于普通 CPU 来说,亦需上百核心的并行手段及上百小时的计算时间才能解决,公开发表的 DNS 结果十分少见。故急需寻找短时间内获得该问题 DNS 结果的手段,以便获得在不同 Re 下的 DNS 结果。由于湍流实验的复杂性,使 DNS 结果对于各种 Re 数下,湍流模型的发展有着不可替代的重要作用。

显卡 GPU 设计的最初目的是极快地处理图形数据,以达到像素渲染效果。近些年来,随着 CUDA,OpenCL 等运算平台的产生与发展,GPU 越来越多地被应用于图像计算以外的领域,这就是所谓的 GPU 通用计算 GPGPU (general-purpose graphics processing unit)。GPGPU 的探索始于 2003 年。随着 Nvidia 公司 CUDA^[7]的发布,针对图形显示处理器 GPU 的编程越来越方便,GPU 已不再仅被用于图像的显示。由于其高超的浮点运算性能(最新的 Tesla Kepler 系列型单个 GPU 有 2 688 个计算核心,单精度理论峰值 3.95TFlops)、良好的编程性(仅简单扩展 C 语言)以及低性价比(Tesla K20 每个约 2.5 万元),人们开始在各行各业中广泛应用,其中工程上基于网格的计算流体力学是其重要的应用之一。根据经验,描述粘性不可压缩流体运动的 Navier-Stokes 方程,采用 MAC(marker and cell)求解器在单个 GPU 上可加速 30~40 倍,并行性好的格子 Boltzmann 方法(LBM)甚至可加速达 100 倍左右^[8-11]。东京工业大学高性能计算中心的青木研究室从 2006 年就开始致力于 GPU 高性能计算,小组成员用上百甚至上千个 GPU 并行,数值模拟了天气预报、晶体生长过程、城市中的空气流动以及在医学上的应用,实现了每秒 $10^{12} \sim 10^{15}$ 次的浮点运算^[12-15]。目前主要关于 GPU 的研究都在计算机领域内致力于如何进一步提高 GPU 的计算性能,在能源、环境、化工、航空航天及医疗等领域,真正将 GPU 作为工程研究手段的还不多。

本研究将 LBM 算法应用于多 GPU 上,对 $Re_\tau = 180$ 的槽道湍流进行直接数值模拟,旨在验证 LBM 模拟湍流的能力和有效性及其 GPU 的加速性能。

1 格子 Boltzmann 方程

格子 Boltzmann 方程是 Boltzmann-BGK 方程的一种特殊离散形式,这一离散包括速度离散、时间离散和空间离散。速度离散的 Boltzmann 方程为

$$\frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i = -\frac{1}{\lambda}(f_i - f_i^{\text{eq}}), \quad (1)$$

其中, f_i 与 f_i^{eq} 分别为离散速度空间的局部分布函数和局部平衡态分布函数, λ 为弛豫时间, \mathbf{e}_i 为有限维速度空间中第 i 方向上的粒子速度, $i = 0, 1, 2, \dots, N-1$, N 为速度空间总维数。若 N 太小,即离散速度的个数太少,可能导致某些应当守恒的物理量不满足守恒定律,若 N 太大则可能造成计算上的浪费。对于二维模型,通常取 $N = 9$,记作 D2Q9 模型。对于三维,通常有 D3Q13, D3Q15, D3Q19, D3Q27 ($N = 13, 15, 19, 27$)。以上几种模型,通常选择局部平衡态分布函数如下:

$$f_i^{\text{eq}} = \rho \omega_i \left[1 + 3\mathbf{e}_i \cdot \mathbf{u} + \frac{9}{2}(\mathbf{e}_i \cdot \mathbf{u})^2 - \frac{3}{2}\mathbf{u} \cdot \mathbf{u} \right], \quad (2)$$

其中, ω 为权系数. 在本研究中, 选择 D3Q19 模型, 如图 1. 相应的权系数为 $\omega_0 = 1/3$, $\omega_1 \sim \omega_6 = 1/18$, $\omega_7 \sim \omega_{18} = 1/36$. ρ 和 \mathbf{u} 分别为宏观密度与速度, 按照下式计算:

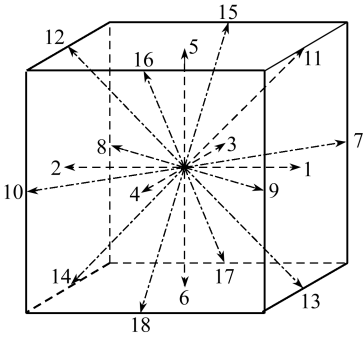


图 1 D3Q19-LBM 模型

Fig. 1 D3Q19-LBM model

$$\begin{cases} \rho = \sum_{i=0}^N f_i = \sum_{i=0}^N f_i^{\text{eq}}, \\ \rho \mathbf{u} = \sum_{i=0}^N f_i \mathbf{e}_i = \sum_{i=0}^N f_i^{\text{eq}} \mathbf{e}_i. \end{cases} \quad (3)$$

LBM 中, 音速 $c_s = 1/\sqrt{3}$, 根据理想气体状态方程可得压力

$$p = \rho c_s^2 = \rho/3. \quad (4)$$

式(1)可继续在空间 x 及时间 t 上离散, 得

$$\begin{aligned} f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = \\ - \frac{1}{\tau} [f_i(\mathbf{x}, t) - f_i^{\text{eq}}(\mathbf{x}, t)], \end{aligned} \quad (5)$$

式(5)称为 LBGK 模型. 其中, $\tau = \lambda/\Delta t$ 为无量纲弛豫时间, 宏观粘度按照下式计算:

$$\nu = \left(\tau - \frac{1}{2} \right) \Delta t. \quad (6)$$

通常, 式(5)按照下面两步骤求解, 并假定 $\Delta t = 1$,

$$\text{碰撞} \quad \bar{f}_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) - \frac{1}{\tau} [f_i(\mathbf{x}, t) - f_i^{\text{eq}}(\mathbf{x}, t)]; \quad (7)$$

$$\text{移流} \quad f_i(\mathbf{x} + \mathbf{e}_i, t + 1) = \bar{f}_i(\mathbf{x}, t), \quad (8)$$

f_i 和 \bar{f}_i 分别代表碰撞前后的分布函数.

对于边界条件, 本文采用如下方法:

$$\bar{f}_i = f_i^{\text{eq}} + f_i^{\text{neq}} = f_i^{\text{eq}} + f_{i_neighbor}^{\text{neq}} = f_i^{\text{eq}} + (\bar{f}_{i_neighbor} - f_{i_neighbor}^{\text{eq}}), \quad (9)$$

f_i^{eq} 按照边界的宏观物理量 \mathbf{u} 和 ρ 计算(式(2)), 而 f_i^{neq} 不能直接计算, 假设其等于邻点的值 $f_{i_neighbor}^{\text{neq}} \cdot \bar{f}_{i_neighbor}$ 和 $f_{i_neighbor}^{\text{eq}}$ 分别按照式(7)与式(2)计算.

对于外力项, LBE 有很多种实施方法, 本文采用修正速度法^[16], 即在求 f_i^{eq} (式(2))之前, 将宏观速度按下式修正:

$$\rho \mathbf{u}^{\text{eq}} = \rho \mathbf{u} + \tau \mathbf{G}, \quad (10)$$

其中, \mathbf{G} 为外力, 用 \mathbf{u}^{eq} 取代 \mathbf{u} 之后代入式(2) 求平衡态分布函数 f_i^{eq} .

2 基于 NS 方程算法与格子 Boltzmann 方法的 GPU 加速性能比较

本文详细比较了 NSE 与 LBM 的 GPU 加速情况. 以二维圆柱绕流为算例, 网格数 1024×512 , GPU 采用 GeForce GTX280 (single), CPU 采用 Intel Xeon E5420 (2.5 GHz). 其中求解 NSE 采用 Red-Black 法及多重网格技术. 表 1 和表 2 分别显示了两种算法的计算结果.

从表 1 可以看出, 整个问题用 GPU 求解加速 13.7 倍(相对于 CPU), 其中加速最慢的部分是求解压力的 Poisson 方程部分. 在 GPU 求解器中, 该部分占总耗时的 82%, 而在相应的 CPU 求解器中, 占 57%. 这同时也说明了, 在开发基于 NS 方程的 GPU 求解器时, Poisson 方程的并行设计是整个 GPU 计算的关键, 其决定了最终的求解器性能. 而 advection-diffusion 部分, GPU 求解器中, 该部分耗时仅占总时间的 8.8%, 而相应的 CPU 求解器为 36%, 这说明, 该部分是

适合于 GPU 并行计算的。

从表 2 可以看出, 对于 LBM, 整个问题用 GPU 求解加速 87.4 倍。其中, 完全本地化的碰撞步(式(7))加速达 124.2 倍。而无计算仅内存读取的移流步(式(8))亦可加速 62.9 倍。在表 2 中, 碰撞步中的 4.28+1.87 表示碰撞步耗时 4.28 s, 边界条件处理部分(非本地化)耗时 1.87 s。

通过上面的比较可以看出, LBM 在 GPU 加速上有着无可比拟的优势, 另外从编程上来讲, LBM-GPU 程序也相对简单得多。

表 1 NSE 模拟二维圆柱绕流求解各项时间、性能及加速比

Table 1 Elapsed time, performance and speed-up of CPU & GPU on simulating 2D flow around a cylinder by NSE

1 024×512	CPU		GPU		speed-up S
	elapsed time	performance	elapsed time	performance	
	<i>t</i> /ms	<i>P</i> [GFlops]	<i>t</i> /ms	<i>P</i> [GFlops]	
overall	282.12	1.24	20.65	16.89	13.7
advection-diffusion	102.43 (36%)	1.15	1.83 (8.8%)	64.23	56.0
divergence <i>U</i>	3.97	0.79	0.18	17.14	21.7
Poisson	160.41 (57%)	1.32	16.94 (82%)	12.49	9.5
gradient <i>p</i>	6.48	0.65	0.26	16.03	24.8

表 2 LBE 模拟二维圆柱绕流求解各项时间、性能及加速比

Table 2 Elapsed time, performance and speed-up of CPU & GPU on simulating 2D flow around a cylinder by LBE

1 024×512	CPU		GPU		speed-up S
	elapsed time	performance	elapsed time	performance	
	<i>t</i> /s	<i>P</i> [GFlops]	<i>t</i> /s	<i>P</i> [GFlops]	
overall	1 345.7	0.57	15.40	50.34	87.4
collision (including BC)	763.6	0.88	6.15 (4.28+1.87)	109.10	124.2
streaming (including macro value computation)	582.1	0.18	9.25	11.30	62.9

3 物理问题模型及参数

图 2 为平板间流动示意图。 x, y, z 分别为流向、法向与展向, 对应的计算区域长度为 L_x, L_y 与 L_z 。 δ 为法向高度的一半, 用来定义 Reynolds 数。标准 Reynolds 数定义为 $Re_c = \delta u_c / \nu$, u_c 为中心面上流速。另外, 对于该问题, 多采用壁面单位系统, 即粘性长度 $l_\tau = \nu / u_\tau$, 摩擦速度 $u_\tau = \sqrt{G\delta/\rho}$, 相应的时间 $t_\tau = l_\tau / u_\tau$ 。由此可得摩擦 Reynolds 数 $Re_\tau = \delta u_\tau / \nu$ 。若用粘性长度表示: $Re_\tau = \delta / l_\tau = \delta^+$, 其中, 上标 (+) 表示壁面单位系统下的无量纲物理量。对于充分发展的平板间湍流, $Re_c \gg Re_\tau$ 。

平板间湍流的直接模拟是一个极其漫长的过程, 为了能节省计算时间, Jimenez 等^[17] 研究了该问题的最小求解区域, 即该区域大小既能维持湍流, 同时对于各个 Reynolds 数下也能够得出正确的结果。定义 $a_x = L_x / \delta, a_z = L_z / \delta$ 。根据文献[17]的研究, Spasov 等^[18] 用 LBM 直接模拟 $Re_\tau = 180$ 下的平板间湍流时, 取 $a_x = 4, a_z = 1$ 。由计算结果来看, 平均流速比 Moser 等的结果^[5] 偏高, Reynolds 应力等吻合得也不是很好。本文中采用 $a_x = 8, a_z = 2$ 。

对于 DNS 来说, 网格尺寸 Δ 不能超过当地 Kolmogorov 长度 η 。对于充分发展的湍流, 在平板边界处 $\eta^+ = 1.5$ ^[19], 离开边界越远, η^+ 越大。由于 LBM 全场采用均一网格, 故全场应满足

$\Delta^+ < \eta^+ = 1.5$. 对于 LBM, $\Delta = 1, \Delta^+ = \Delta/l_\tau = Re_\tau/\delta < 1.5$, 可得 $\delta > Re_\tau/1.5$. 若 $Re_\tau = 180$, 则有 $\delta > 120$. 本文计算工况采用 $Re_\tau = 180$, 根据以上推导, 网格数 $L_x \times L_y \times L_z$ 为: $1\ 024 \times 256 \times 256$. 由此可得 $\Delta^+ = 1.41$.

本计算中的初始条件: u_τ 按照对数摩擦定律给出, 即 $u_c/u_\tau = (1/\kappa) \ln Re_\tau + b$, 其中 κ 为 von Kármán 常数, $\kappa \approx 0.4$, $b \approx 6$ (由实验得出). u_c 为中心面速度, 为输入参数, $u_c = 0.1$. 其次, 有 $u = \bar{u} + u', v = \bar{v} + v', w = \bar{w} + w', \rho = \rho_0 = 1$. 其中, $u' = v' = w' = u_c(r_{\text{rand}} - 0.5)$, $r_{\text{rand}} \in [0, 1]$, $\bar{v} = \bar{w} = 0$, \bar{u} 按下式计算:

$$u^+ = \begin{cases} y^+, & y^+ \leq y_w, \\ \ln(y^+)/\kappa + b, & y^+ > y_w, \end{cases} \quad (11)$$

u^+ 为壁面单位系统下无量纲速度, $u^+ = \bar{u}/u_\tau, y_w = 11.6$.

本计算中的边界条件: 进出口及壁面处均按照式(9)计算, z 方向为周期性边界条件. 根据文献[16], 平均统计的时间至少为充分发展后 $5t_t, t_t = a_x \delta / u_\tau$. 根据本文参数, $5t_t$ 即第 2×10^6 至 3×10^6 个 LBM 时间步长. 图 3 为本计算中中心面上平均速度随时间变化图, 图上标明了本计算中平均统计的时间.

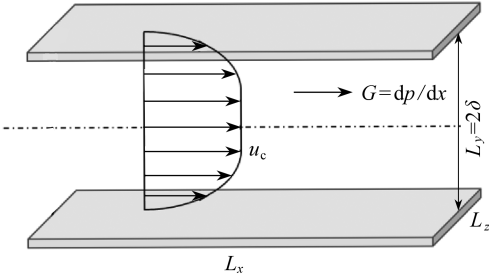


图 2 物理模型

Fig. 2 Calculation model for flow between plates

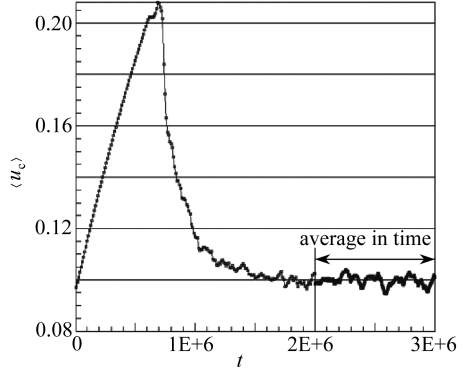


图 3 中心面上平均速度随计算时间变化图

Fig. 3 Profile of average velocity with time at the central plane

4 LBM 的多 GPU 实施

格子 Boltzmann 法的天生并行性是被广大研究人员所熟知的, 基本的 LBM 求解器在单个 GPU 上加速可达 130 倍左右. 然而, LBM 的缺点之一是变量数太多, 极耗内存. 对于 D3Q19 模型单精度计算, 理论上显存为 3 G 的 GPU 能处理的网格数约为 10^7 , 对于本计算的网格数 6.7×10^7 , 至少需要 7 个 GPU. 另外, GPU 之间无法独立进行数据通讯, 多 GPU 之间的数据通讯是通过 CPU 完成的. 关于 LBM 单 GPU 与多 GPU 并行的实施方法, 详见文献[11]. 本计算中采用 8 个 NVIDIA Tesla M2050 GPU, 用 MPI (message passing interface) 及 CUDA 中 CudaMemcpy() 语句实现 GPU 间的数据传输, 单精度计算. 对多 GPU/CPU 的并行, 据以往的经验^[11], 在 GPU 数目不超过 10 个的情况下, 一维区域分割法最为经济. 另外, 一维分割时, 通讯量不随进程数的改变而改变. 对本文中的问题, 虽然沿 x 方向上区域分割所产生的通讯量为沿 y, z 方向上分割的通讯量的 1/4, 为了便于得到沿法向 (y) 上各横截面上 ($x-z$) 的平均速度, 本计算在并行上采用沿法向 (y) 的一维区域分割, 每个 GPU 上的计算量 (网格数) 为 $1\ 024 \times 32 \times 256$, 如图 4 所示. 并行计算采用一维区域分割时, D3Q19 模型所传递的变量详见文献[11].

另外,需要指出的是,采用 CPU 的 LBM 求解器时,式(7)和式(8)在顺序上并无先后之分,而采用 GPU-LBM 时,先计算移流项(式(8))可以大大减少访问显存的次数,使得整体效率提高 30% 左右.

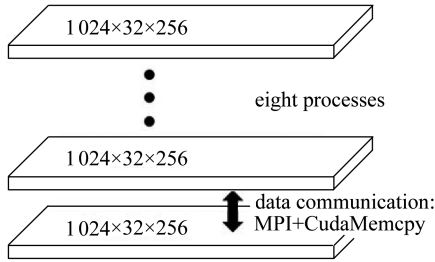


图 4 一维区域分割示意图

Fig. 4 One-dimensional domain decomposition

5 DNS 结果

本次计算 3×10^6 个 LBM 时间步长,8 个 GPU,用时约 24 h. 其性能为每秒处理 2.33×10^9 网格. 同样地直接模拟,文献[18]中采用 36 个 CPU 并行,性能仅为每秒处理 5.60×10^6 网格. 本文的计算性能为文献[18]中的 416 倍.

图 5 为速度梯度张量第二不变量的等值面图,是流场内涡的综合体现.

图 6 为 $x-z$ 截面平均速度 $\langle u(y) \rangle$ 沿法向的变化曲线,实线为本文 DNS 结果,虚线所示为 Moser 等的 DNS 结果^[5]. 由图 6 可以看出,二者吻合得非常完美. 在此之前,本文也计算了 $a_x = 4, a_z = 1$ 及 $a_x = 8, a_z = 1$ 的工况,结果表明,在靠近中心线处,平均速度略高于 Moser 的结果. 而 $a_x = 4, a_z = 2$ 的结果却吻合得很好. 这说明,虽然在展向 z 上采用周期性边界条件,该方向上的计算长度也会对结果产生较大影响.

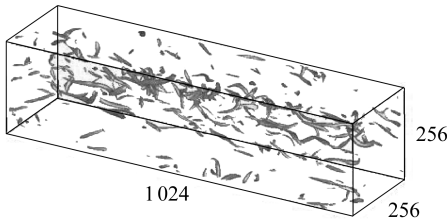


图 5 速度梯度张量第二不变量的等值面图

Fig. 5 Iso-surface of second invariant of velocity gradient tensor

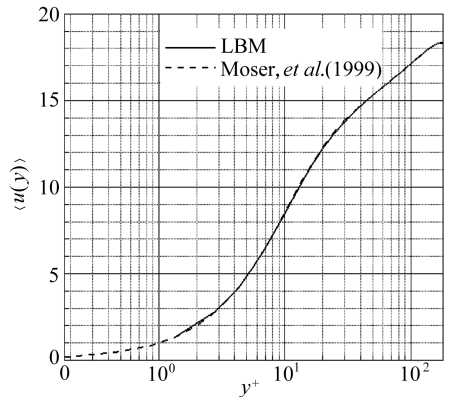
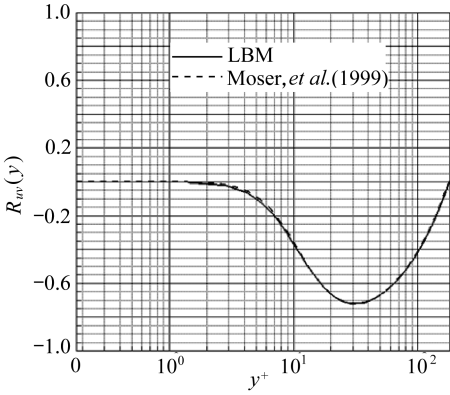


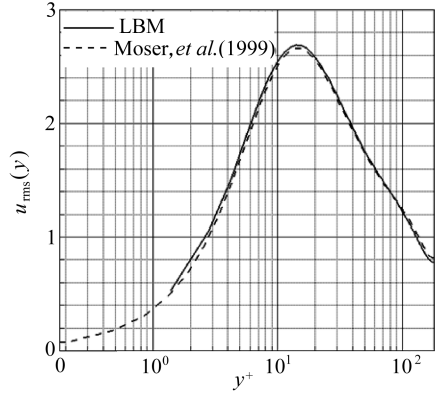
图 6 平均速度在法向 y 上的变化曲线

Fig. 6 Profile of average velocity in the normal y -direction

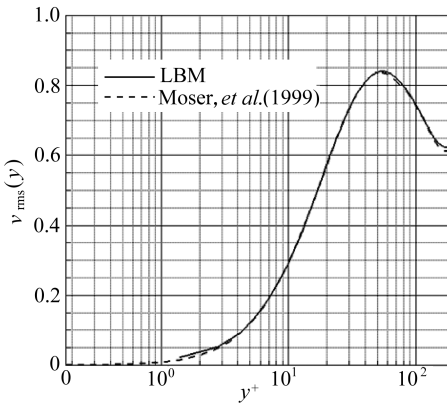
图 7 为湍流统计量: Reynolds 应力 $R_{uv}(y)$ (图 7(a)) 及脉动速度均方根 $u_{i,rms}(y)$ (图 7(b) ~ (d)) 沿法向的变化曲线,且与 Moser 等的 DNS 结果^[5] 作对比. 从图 7 可以看出,各个量都吻合得非常好. 同样地,用 $a_x = 4, a_z = 1$ 和 $a_x = 8, a_z = 1$ 以及 $a_x = 4, a_z = 2$ 计算,湍流统计量与 Moser 的结果均略有偏差.



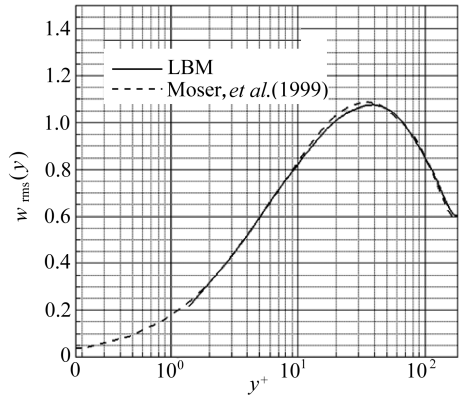
(a) $R_w(y) = u'v'/u_\tau^2$



(b) $u_{rms}(y) = u'/u_\tau$



(c) $v_{rms}(y) = v'/u_\tau$



(d) $w_{rms}(y) = w'/u_\tau$

图7 Reynolds 应力 $R_w(y)$ 及脉动速度均方根 $u_{i,rms}(y)$ 沿法向的变化曲线

Fig. 7 Profiles of Reynolds stress $R_w(y)$ and rms of velocity fluctuations $u_{i,rms}(y)$ in the normal y -direction

6 结 论

本文采用格子 Boltzmann 法对平行板间的湍流做了直接模拟, 得到以下结论:

- 1) DNS 结果显示: 在平均速度及湍流统计量上均与 Moser 等的 DNS 结果吻合得很好, 通过对比, 证实了二阶精度的格子 Boltzmann 方法模拟湍流的有效性;
- 2) 在计算中, 采用了计算能力超高的 GPU 作为计算工具, 并与格子 Boltzmann 法的良好并行性完美匹配, 使得在短时间内获得网格数近亿的 DNS 结果, 即 24 h 内完成 0.67 亿网格的 300 万时间 LBM 步长的计算, 性能达到 2 330MLUPS, 即每秒处理 2.33×10^9 个 LBM 网格;
- 3) 对比了传统的 NSE 与 LBE 在 GPU 上的加速性能. 可以得出: LBM 与 GPU 的完美结合, 使得 CFD 领域的 DNS 不再成为难题, 并为需要高分辨率网格的方法及应用提供了更广阔的发展前景.

参考文献 (References):

[1] Chen S Y, Doolen G D. Lattice Boltzmann method for fluid flows[J]. *Annual Review of Fluid Mechanics*, 1998, **30**: 329-364.

- [2] 何雅玲, 王勇, 李庆. 格子 Boltzmann 方法的理论及应用[M]. 北京: 科学出版社, 2009. (HE Ya-ling, WANG Yong, LI Qing. *Lattice Boltzmann Method: Theory and Applications* [M]. Beijing: Science Press, 2009. (in Chinese))
- [3] Yu H, Girimaji S S, Luo L S. DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method[J]. *Journal of Computational Physics*, 2005, **209**(2): 599-616.
- [4] Yu H, Luo L S, Girimaji S S. LES of turbulent square jet flow using an MRT lattice Boltzmann model[J]. *Computers & Fluids*, 2006, **35**(8/9): 957-965.
- [5] Moser R D, Kim J, Mansour N N. Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$ [J]. *Phys Fluids*, 1999, **11**(4): 943-945.
- [6] Kim J, Moin P, Moser R D. Turbulence statistics in fully developed channel flow at low Reynolds number[J]. *J Fluid Mech*, 1987, **177**: 133-166.
- [7] Nvidia. NVIDIA CUDA Programming Guide[K]. Version 2.0. 2008.
- [8] Ogawa S, Aoki T. GPU computing for 2-dimensional incompressible-flow simulation based on multi-grid method[C]//*Transactions of JSCEs*. Paper No 20090021. 2009.
- [9] Harada T. Smoothed particle hydrodynamics on GPUs[C]//*Proceeding of the Spring Conference on Computer Graphics*, 2007: 235-241.
- [10] Rossinelli D, Bergdorf M, Cottet G-H, Koumoutsakosa P. GPU accelerated simulations of bluff body flows using vortex particle methods[J]. *Journal of Computational Physics*, 2010, **229**(9): 3316-3333.
- [11] Wang X, Aoki T. Multi-GPU performance of incompressible flow computation by lattice Boltzmann method on GPU cluster[J]. *Parallel Computing*, 2011, **37**(9): 521-535.
- [12] Shimokawabe T, Aoki T, Takaki T, Endo T, Yamanaka A, Maruyama N, Nukada A, Matsuo S. Peta-scale phase-field simulation for dendritic solidification on the TSUBAME 2.0 supercomputer[C]//*Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. New York, USA, 2011.
- [13] Shimokawabe T, Aoki T, Ishida J, Kawano K, Muroi C. 145 TFlops performance on 3990 GPUs of TSUBAME 2.0 supercomputer for an operational weather prediction[C]//*Proceedings of the International Conference on Computational Science, ICCS 2011*, 2011, **4**: 1535-1544.
- [14] Wang X, Aoki T. High performance computation by multi-node GPU cluster-TSUBAME 2.0 on the air flow in an urban city using lattice Boltzmann method[J]. *International Journal of Aerospace and Lightweight Structures*, 2012, **2**(1): 77-86.
- [15] Miki T, Wang X, Aoki T, Imai Y, Ishikawa T, Takase K, Yamaguchi T. Patient-specific modeling of pulmonary air flow using GPU cluster for the application in medical practice[J]. *Computer Methods in Biomechanics and Biomedical Engineering*, 2012, **15**(7): 771-778.
- [16] Lammers P, Beronov K N, Volkert R, Brenner G, Durst F. Lattice BGK direct numerical simulation of fully developed turbulence in incompressible plane channel flow[J]. *Computers & Fluids*, 2006, **35**(10): 1137-1153.
- [17] Jimenez J, Moin P. The minimal flow unit in near-wall turbulence[J]. *Journal of Fluid Mechanics*, 1991, **225**(1): 213-240.
- [18] Spasov M, Rempfer D, Mokhasi P. Simulation of turbulent channel flow with an entropic lattice Boltzmann method[J]. *Int J Numer Meth Fluids*, 2009, **60**(11): 1240-1258.
- [19] Pope S B. *Turbulent Flows*[M]. Cambridge: Cambridge University Press, 2000.

Direct Numerical Simulation of the Wall-Bounded Turbulent Flow by Lattice Boltzmann Method Based on Multi-GPU

XU Ding, CHEN Gang, WANG Xian, LI Yue-ming

(State Key Laboratory for Strength and Vibration of Mechanical Structures, School of Aerospace, Xi'an Jiaotong University, Xi'an 710049, P. R. China)

Abstract: The wall-bounded turbulent flow was simulated directly (DNS) by lattice Boltzmann method (LBM) through multi-GPU parallel computing. The Data-parallel SIMT (single-instruction multiple-thread) characteristic of GPU matched the parallelism of LBM well, which led to high efficiency of GPU on the LBM solver. At the same time, it brought possibility for large-scale DNS on the desk-top supercomputer. In this DNS work, 8 GPUs were adopted. The number of meshes of 6.7×10^7 , which resulted in a non-dimensional mesh size of $\Delta^+ = 1.41$ for the whole solution domain. It took only 24 hours for the GPU-LBM solver to simulate 3×10^6 LBM steps. As a result, both the mean velocity and turbulent variables, such as Reynolds stress and velocity fluctuations, agree well with the results of Moser, et al. The capacity and validity of LBM in simulating turbulent flow are verified.

Key words: lattice Boltzmann method; multi-GPU parallel computing; wall-bounded turbulent flow; DNS

Foundation item: The National Natural Science Foundation of China(11242010;11102150)