有限体积格子 Boltzmann 方法的 算法改进及性能分析^{*}

武 频,曹啸鹏,尚伟烈,郑德群,高 升 (上海大学 计算机工程与科学学院,上海 200072)

摘 要:有限体积格子 Boltzmann 方法(LBM)能够将标准 LBM 的应用范围扩展到非结构网格,但是比起标准的 LBM 这个方法需要更多的内存用量和计算量。针对此问题采用了优化计算顺序、简化计算方程的方法对有限体 积 LBM 算法进行改进。科学的分析和实验的结果表明,改进后的算法能够在不增加计算量的基础上减少内存 用量,在一些情况下还可以大量减少计算时间。

关键词:有限体积法;格子 Boltzmann;算法改进;内存优化;减少计算

中图分类号: TP391 文献标志码: A 文章编号: 1001-3695(2012)10-3706-04 doi:10.3969/j.issn.1001-3695.2012.10.026

Optimization and performance analysis for code of finite volume lattice Boltzmann method

WU Pin, CAO Xiao-peng, SHANG Wei-lie, ZHENG De-qun, GAO Sheng

(School of Computer Engineering & Science, Shanghai University, Shanghai 200072, China)

Abstract: Finite volume lattice Boltzmann method (LBM) can extend the standard LBM to unstructured mesh, but compared with standard LBM this method suffers from higher memory consumption and poorer computational performance. In order to solve this problem, the improvement process used the methods of optimizing evaluation order and simplifying calculating equation. Scientific analysis and experimental results demonstrate that the improved algorithm results in lower memory usage without additional computation, and in some conditions it reduces much computation.

Key words: finite volume; lattice Boltzmann; algorithm improvement; memory optimizaion; computation reduction

0 引言

格子 Boltzmann 方法(LBM) 从介观的层次对流体流动进 行描述,它通过构造碰撞和迁移的演化机制,让介观层次中的 流体微团进行演化计算,然后通过对微团状态的统计得到宏观 物理参数。这种以简单数学模型对复杂物理系统实现仿真的 机制使 LBM 具有许多常规数值方法所没有的优点,如程序结 构简洁、边界处理方便和易于大规模并行计算等。正是由于具 有这些独特的优势,LBM 吸引了越来越多的学者对其进行深 人研究:a)在物理方面,扩展其适用的流体类型^[1,2];b)在网格 技术方面,少网格限制,提高计算精度^[3-6];c)在计算机技术方 面,减少计算时间和内存使用量^[7-11]。

由于标准的 LBM,即 LBGK 模型^[12],只适用于不可压缩流 动和不太高的雷诺数,为了解决这些问题,学者们提出了适用 于不可压缩流动问题的多能量级模型^[1]和适用于高雷诺数问 题的大涡模拟模型^[2]。另外,此模型在网格方面有固有的缺 点,如网格划分必须对称均匀和计算精度不可调等,因此有限 差分 LBM^[3]、有限体积 LBM^[4,5]和多块网格^[6]的技术被提出 以解决这类问题。LBM 具有天生并行性,所以如非阻塞通 信^[7]、GPU 辅助计算^[8]等并行计算技术被应用以减少计算时间;在减少内存使用量方面,也有迁移算法^[9]、交换算法^[10]和向量存储法^[11]等被相继提出。

但是,目前提出的减少计算时间和内存使用量的方法都是 针对 LBGK 模型,在有限体积 LBM 模型上并没有类似的方法 被提出,本文将提出改进有限体积 LBM 模型的代码优化方法。

1 有限体积 LBM

由于标准的 LBM 只适用于正六边形、正方形等对称均匀 的网格,并且所有网格的形状和大小必须一致,从而使此方法 的应用受到了很多限制。另一方面,以积分守恒方程为基础的 有限体积法适用于非结构网格,而且非结构网格在处理复杂边 界和调节精度方面所受的限制要远小于规则网格。因此,不少 学者将有限体积法的思想融入 LBM 中,以减少其在网格方面 所受到的限制,于是有限体积 LBM 被提出来了。此方法可以 分为两类:a)将速度、密度等物理量存储在控制单元节点上的 格点法,如文献[4]所述;b)将物理量存储于控制单元中心的 方法,如文献[5]所述。本文所描述与改进的对象是后者。

收稿日期: 2012-02-28; 修回日期: 2012-04-01 基金项目: 国家自然科学基金资助项目(11002086);国家科技支撑计划资助项目(2009BAG12A01,2009BAG11B02)

作者简介: 武频(1975-), 女, 河北石家庄人, 副教授, 博士, 主要研究方向为高性能计算、计算流体力学(wupin@ shu. edu. cn); 曹啸鹏(1987-), 男, 河北石家庄人, 硕士, 主要研究方向为高性能计算; 尚伟烈(1975-), 男, 吉林人, 博士研究生, 主要研究方向为高性能计算、计算流体力学; 郑德 群(1987-), 男, 山东兖州人, 硕士, 主要研究方向为高性能计算; 高升(1989-), 男, 山东聊城人, 硕士, 主要研究方向为高性能计算.

}

使用 BGK 近似将 Boltzmann 方程中的碰撞项简化后,再从 速度空间将其离散可得到离散 Boltzmann-BGK 方程:

$$\frac{\partial f_{\alpha}}{\partial t} + e_{\alpha} \times \nabla f_{\alpha} = -\frac{1}{\tau} [f_{\alpha}(x,t) - f_{\alpha}^{eq}(x,t)]$$
(1)

其中:7 是弛豫时间,用于描述粒子分布函数 f_a 从非平衡态向 平衡态趋近的变化过程;f^m_a(x,t)是平衡态分布函数,其具体形 式与离散速度模型的构造有关。本文中的速度离散模型采用 DdQm 系列模型^[12]。根据减少可压缩性误差的 He-Luo 模 型^[13],此系列速度离散模型采用如下形式的平衡态分布函数:

$$f_{\alpha}^{eq} = w_{\alpha} \left[p + p_0 \left(\frac{e_{\alpha} \times u}{c_s^2} + \frac{(e_{\alpha} \times u)^2}{2c_s^4} - \frac{u^2}{2c_s^2} \right) \right]$$
(2)

在这种模型中,流体的宏观压力、宏观速度和宏观密度定 义如下:

$$p(x,t) = \sum f_{\alpha}(x,t) \tag{3}$$

$$u(x,t) = \sum f_{\alpha}(x,t) e_{\alpha}/p_0 \tag{4}$$

$$\rho(x,t) = p(x,t)/c_s^2 \tag{5}$$

以二维问题为例,假设流体区域被离散成如图1所示的拥 有任意形状和尺寸的三角形单元。

在离散单元内对式(1)左右两侧求积分并化简后,可得到 如下方程:

$$\frac{\mathrm{d}}{\mathrm{d}t}(f_{\alpha,i}) = L(f_{\alpha,i}) = L_{\alpha,i}^{C} + L_{\alpha,i}^{A}$$
(6)

$$L^{A}_{\alpha,i} = -\frac{1}{A_{i}} \oint_{\partial \Omega_{i}} (e_{\alpha} \times n) f_{\alpha} dl = -\frac{1}{A_{i}} \sum_{m} \overline{F}(f_{\alpha,im}) \Delta l_{im}$$
(7)

$$L_{\alpha,i}^{\mathcal{C}} = -\frac{1}{A_{i}\tau} \int_{\Omega_{i}} \left[f_{\alpha}(x,t) - f_{\alpha}^{eq}(x,t) \right] \mathrm{d}\Omega = -\frac{1}{\tau} (f_{\alpha,i} - f_{\alpha,i}^{eq}) \quad (8)$$

其中: L^A 和 L^c 分别代表对流通量和碰撞项; Ω_i 表示任一单元 ABC,其边界和面积分别是 $\partial\Omega_i$ 和 A_i ,单元的每条边及其边长分 别使用 l_{im} 和 Δl_{im} ($m \in \{i, j, k\}$)表示。本文中时间的离散方式 采用显示的一阶前向差分,在 $t + \Delta t$ 时刻分布函数的值可由式 (9)求出:

$$f_{\alpha,i}^{t+\Delta t} = f_{\alpha,i}^{t} + \Delta t L(f_{\alpha,i}^{t})$$
(9)

单元边上的数值通量可以采用如下方法进行计算:

$$\overline{F}(f_{\alpha,im}) = f_{\alpha,i*} \times e_{\alpha} \times n_{im}$$
(10)

$$f_{\alpha,i*} = \begin{cases} f_{\alpha,i} + \frac{1}{2} \varPhi_{im} (f_{\alpha,m} - f_{\alpha,i}) & e_{\alpha} \times n_{im} > 0 \\ \\ f_{\alpha,m} + \frac{1}{2} \varPhi_{im} (f_{\alpha,i} - f_{\alpha,m}) & e_{\alpha} \times n_{im} \leqslant 0 \end{cases}$$
(11)

其中: n_{in} 是垂直于 $\partial \Omega_i$ 且指向 Ω_i 外侧的单位向量; i^* 代表边 l_{im} 的迎风质心。有限体积 LBM 中豫弛时间的计算方法与标准的 LBM 不同,计算方法如下:

$$\tau = v/c_s^2 \tag{12}$$

本文中使用的边界处理方式为非平衡反弹格式[5,14]。

2 LBM 计算方法的改进

2.1 原算法的计算方法

根据式(6)可知有限体积 LBM 的核心计算过程可分为计 算对流通量和计算碰撞项两部分。对于任意单元的对流通量, 其计算过程又可以分为三步:a)对于任意一个粒子分布函数 计算出它在一条边上的数值通量;b)将每条边的数值通量乘 以 Δ*l_{im}/A_i* 后求和,这时得到的是这个粒子分布函数的对流通 量;c)对一个单元中的所有粒子分布函数执行前两步。

为了完成这些计算过程,需要引入两个函数:a)根据式

(10)和(11)计算数值通量的函数 Compute_Advective,它的参数分别用于索引单元信息、单元中的分布函数和单元中边的信息;b)根据式(8)计算单元中分布函数的碰撞项的函数 Compute_Collision。假设一个单元有 *s* 条边和 *q* 个分布函数,这样核心计算过程可用伪代码表示如下:

$$\begin{aligned} & \text{for}(\alpha = 0; \alpha < q; \alpha ++) \\ & \text{for}(m = 0; m < s; m ++) \\ & \text{L}_{advective} + = \\ & \text{Compute}_Advective(i, \alpha, edge_{im}) \times \Delta l_{im} / A_i; \} \\ & \text{for}(\alpha = 0; \alpha < q; \alpha ++) \\ & \text{f}_{next} = f + \end{aligned}$$

 $\Delta t \cdot (Compute_Collision(i, \alpha) + L_{advective}); \}$

在此计算过程中所有单元是逐个独立演进的,这样会引入 两处额外的内存使用:a)在计算当前单元的边上的数值通量 时需要依靠相邻单元中的粒子分布函数,并且由时间离散方式 可知这些粒子分布函数都必须处于同一时间层,所以,为了保 证原粒子分布函数不被覆盖,需要将新计算出的粒子分布函数 与原粒子分布函数分别存储;b)在计算对流通量时每条边被 包含它的单元各使用了一次,而边长、单位法向量和邻接单元 等边的信息是包含于单元信息中的,这样造成了边信息的重复 存储。

2.2 改进的计算方法

以前相邻的两个单元各自计算关于它们临边的数值通量, 现在这些计算将交由其中一个单元执行。假设每个单元需要 计算 s。条边上的数值通量,则改进方法的伪代码如下:

$$for(\alpha = 0; \alpha < q; \alpha ++)$$

$$for(m = 0; m < s_{n}; m ++)$$

$$L_{advective}^{i}[\alpha] + =$$

$$Compute_Advective(i, \alpha, edge_{im}) \times \Delta l_{im}/A_{i};$$

$$L_{advective}^{m}[\alpha] + =$$

$$Compute_Advective(m, \alpha, edge_{im}) \times \Delta l_{im}/A_{m};$$

$$for(\alpha = 0; \alpha < q; \alpha ++)$$

$$f = f +$$

=1+

 $\Delta t \times (Compute_Collision(i, \alpha) + L^{i}_{advective}[\alpha]); \quad \}$

这样既可以保证两个相邻单元的粒子分布函数在被用于 计算数值通量时处于同一时间层,又可以使被相邻单元重复使 用和存储的边信息现在只需被使用和存储于一个单元中。所 以在信息存储方面,新算法可以只存储一个时间层的粒子分布 函数,并且边信息的存储量也可以减至原来的一半。然而新算 法也需要增加一部分存储结构。在原算法中,任意单元中所有 粒子分布函数的对流项在程序处理此单元时可以完整地被计 算得出,因此如原算法的伪代码所示只需要一个临时变量表示 对流项即可。在新算法中,单元中粒子分布函数的对流通量既 有相邻单元计算出的部分,又有本单元计算出的部分,为了保 证先计算出的数据不丢失,需要把对流项的存储结构加入到每 个单元的数据结构中。具体的存储结构将在本文第3章被详 细对比。

原算法在求任意单元的对流通量时都需要执行 *s* × *q* 次函数 Compute_Advective。尽管在新算法中计算每条边的数值通量时需要多执行一次 Compute_Advective,但是由于被重复使用的边信息现在只需要使用一次,所以边数 *s*_n 平均到每个单元后只相当于原算法的一半,这样在计算量方面新算法与原算法

是持平的。

然而通过进一步的简化方程可以减少计算量。首先,由于 在 DdQm 离散速度模型中 $e_0 = (0,0)$,因此 $\overline{F}(f_{0,im}) \equiv 0$,这样 e_0 处的对流通量计算可以省去。其次,通过建立 $\overline{F}(f_{\alpha,ji})$ 与 $\overline{F}(f_{\alpha,ji})之间的简单映射关系,能够减少函数 Compute_Advec$ tive 的使用次数。如图 2 所示,对于单元 <math>i,如果有 $e_{\alpha} \times n_{ij} > 0$, 则 α 处的粒子分布函数相对于边 ℓ_m 的数值通量可以表示为

$$\overline{F}(f_{\alpha,ij}) = \left[f_{\alpha,i} + \frac{1}{2}\boldsymbol{\Phi}_{ij}(f_{\alpha,j} - f_{\alpha,i})\right] \times e_{\alpha} \times n_{ij}$$
(13)

其中: Φ 由 i^* 处分布函数的重构方式^[5,15]决定。另一方面,对 于单元j会有 $e_{\alpha} \times n_{j} < 0$,所以:

$$\overline{F}(f_{\alpha,ji}) = \left[f_{\alpha,i} + \frac{1}{2} \Phi_{ji}(f_{\alpha,j} - f_{\alpha,i})\right] \times e_{\alpha} \times n_{ji}$$
(14)



 $T^{+}(12)$ $T^{+}(14)$ $T^{+}(12)$ $T^{+}(14)$ $T^{+}(14)$

使用式(13)减去(14),并利用 $F(f_{\alpha,i})$ 去表示 $F(f_{\alpha,i})$ 能够 得到

$$\overline{F}(f_{\alpha,ji}) = \frac{\frac{1}{2}\overline{F}(f_{\alpha,ij})\Delta\Phi - \overline{F}(f_{\alpha,ij})}{1 - \frac{1}{2}\Delta\Phi}$$
(15)

其中: $\Delta \Phi = \Phi_{ij} - \Phi_{ji}$ 。如果使用片段常数法^[5,15]作为重构方 式,则 $\Phi_{ij} = \Phi_{ji} = 常数,进而式(15)可以化简为<math>F(f_{\alpha,ii}) = -F$ ($f_{\alpha,ij}$)。所以原来需要被执行两次的计算数值通量的函数 Compute_Advective可以只被执行一次,另一次计算可以被取反 操作所代替。具体的伪代码表述如下:

$$\begin{aligned} & \text{for}(\alpha = 1; \alpha < q; \alpha ++) \\ & \text{for}(m = 0; m < s_n; m ++) \\ & \text{for}(m = 0; m < s_n; m ++) \\ & \text{functions}[\alpha] + = \\ & \text{Compute}_Advective(i, \alpha, edge_{im}) * \Delta l_{im}; \\ & L^m_{advective}[\alpha] + = -L^i_{advective}[\alpha]; & \text{for}(\alpha = 0; \alpha < q; \alpha ++) \\ & \text{for}(\alpha = 0; \alpha < q; \alpha ++) \\ & \text{for}(\alpha = 0; \alpha < q; \alpha ++) \end{aligned}$$

 $\Delta t * (Compute_Collision(i, \alpha) + L_{advective}^{1} \lfloor \alpha \rfloor / A_{i});$

第二种深入改进可以减少近一半的对流通量计算,但是也 把分布函数的重构方式限定在片段常数插值和片段线性插 值等^[5,15]。

3 性能分析及实验

对于每个单元的数据存储结构,有四部分信息需要被存储 在每个单元里面,分别是单元的几何信息、边的几何信息、粒子 分布函数以及宏观物理量。以 D2Q9 离散速度模型为例,单元 的数据结构及其所需的物理空间如表1所示,其中数据所占用 空间的单位是机器字。

在原算法中所需要计算的边数为 s, 而在新算法中这个数 目可以看做是 s/2。尽管新算法需要额外的空间用于存储对 流项数据,但是通过省去存储当前时间步的粒子分布函数, 同 样大小的空间可以被节约。另外, 由于调用函数 Compute_Ad-

表1 单元的数据结构

数据	数据约	数据类型		总占用空间					
单元的几何信息	中心坐标 边界类型 面积 边数	double int l double int sid	double x,y int bc double area int sides		8				
边的几何信息	长度 单位法向量 相邻单元	double l double n_ int neig	double length double n_x,n_y int neighbor		7×边数				
粒子分布函数	当前时间步(原算) 下一时间步 对流项(新算法)	去) double double fn) double fad	;) double f[9] double fnext[9] double fadvec[9]		36				
宏观物理参量	压力 密度 速度	doubl double double t	double p double rho double ux,uy		8				
表 2 内存用量比较									
计算用例	离散速度模型 和网格类型	网格数量	M ₀ /1	Byte	M_1 /Byte				
二维方腔	D2Q9 三角形 D2Q9 四边形	22 000 10 000	1. 3754 1. 3357	e + 08 1. e + 08 1.	3663 e + 08 3335 e + 08				
三维方腔	D3Q19 四面体 D3Q19 六面体	107 867 27 000	2. 1528 1. 5726	e + 08 1. e + 08 1.	8736e + 08 4617e + 08				
	D200 三角形	27 442	1 3031	o ±08 1	3803 + 08				

计算用例选取雷诺数为1000 流场尺寸分别100×100和30×30×30的顶盖驱动方腔流以及雷诺数为20流场尺寸为250×50 的圆柱绕流。改进算法和原算法均使用片段常数插值方法,这种方法 $\Phi \equiv 0$ 。网格的生成选用 Gambit 软件包,程序运行的软件环境为 Microsoft Visual Studio 2008,硬件环境为IBM HS21 刀片服务器。每个算例的迭代次数为50000。新算法与原算法在计算时间上的对比如表3所示。设原算法的计算时间为 T_0 ,新算法的计算时间为 T_1 。

12 321

1.3479e + 08 1.3391e + 08

表3 计算时间比较

计算用例	离散速度模型 和网格类型	网格数量	T_0 /s	T_1/s	
二维方腔	D2Q9 三角形	22 000	1 654.00	1 117.41	
	D2Q9 四边形	10 000	587.32	435.45	
三维方腔	D3Q19 四面体	107 867	22 145.03	14 229.77	
	D3Q19 六面体	27 000	4 448.46	3 439. 43	
圆柱绕流	D2Q9 三角形	27 442	2 095.75	1 426. 19	
	D209 四边形	12 321	827.84	588.30	

在文中列举的几种常用的离散速度模型和网格模型上新 算法都取得了不错的实验效果,但是新算法在其他离散速度模 型和网格模型上也能降低内存使用量和减少计算时间,这是由 于新算法是通过减少每个单元所需要存储和使用的边实现优 化效果的,所以无论是什么计算模型,只要存储相互邻接的单 元新算法都可以改善内存使用量以及计算时间。

4 结束语

圆柱绕流

D2Q9 四边形

通过本文的分析和实验可以确定:新算法在离散速度模型 和网格模型方面具有广泛的适用性,并且在降低内存使用量和 减少计算时间方面能取得令人满意的效果。但是计算时间的大 量减少是以特定的数值通量计算方法为条件的,如片段常数插 值和片段线性插值等。对于其他计算方法如移动最小二乘^[16]、 最大压缩和最大扩散等,目前还只是可以降低内存使用量,但无 法节省大量的计算时间。扩展新方法以使其可以在更多的重构 方式下减少计算时间是未来的研究重点。

参考文献:

- YAN Guang-wu, ZHANG Jian-ying, LIU Yan-hong, et al. A multienergy-level lattice Boltzmann model for the compressible Navier-Stokes equations[J]. International Journal for Numerical Methods in Fluids,2007,55(1):41-56.
- [2] HOU S, STERLING J, CHEN S, et al. A lattice Boltzmann subgrid model for high reynolds number flows[J]. Fields Institute Communications, 1994, 6:151-166.
- [3] WANG Yong, HE Ya-ling, ZHAO Tian-shou, et al. Implicit-explicit finite-difference lattice Boltzmann method for compressible flows[J]. International Journal of Modern Physics C,2007,18(12):1961-1983.
- [4] PENG Gong-wei, XI Hao-wen, DUNCAN C. Finite volume scheme for the lattice Boltzmann method on unstructured meshes[J]. Physical Review E,1999,59(4):4675-4682.
- [5] PATIL D V, LAKSHMISHA K N. Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh [J]. Journal of Computational Physics,2009,228(14):5262-5279.
- [6] OLGA F, DIETER H. Grid refinement for lattice-BGK models[J]. Journal of Computational Physics, 1998, 147(1):219-228.
- [7] CLAUSERN J R, Jr REASOR D A, AIDUN C K. A parallel performance of a lattice-Boltzmann/finite element cellular blood flow solver on the IBM Blue Gene/P architecture[J]. Computer Physics Communications, 2010, 181 (6) :1013-1020.

- [8] FREDERIC K, CHRISTIAN O, GILLES R, et al. LBM based flow simulation using GPU computing processor [J]. Computers and Mathematics with Applications, 2010,59(7):2380-2392.
- [9] POHL T, KOWARSCHIK M, WILKE J, et al. Optimization and profiling of the cache performance of parallel lattice Boltzmann codes [J]. Parallel Process Letters, 2003, 13(4):549-560.
- [10] MATTILA K, HYVULUOMA J, ROSSI T, et al. An efficient swap algorithm for the lattice Boltzmann method [J]. Computer Physics Communications, 2007, 176(3):200-210.
- [11] VIDAL D, ROY R, BERTRAND F. A parallel workload balanced and memory efficient lattice Boltzmann algorithm with single unit BGK relaxation time for laminar Newtonian flows [J]. Computers & Fluids,2010,39(8):1411-1423.
- [12] QIAN Y H, D'HUMIERES D, LALLEMAND P. Lattice BGK models for Navier-Stokes equation [J]. Europhysics Letters, 1992, 17(6):479-484.
- [13] HE Xiao-yi, LUO Li-shi. Lattice Boltzmann model for the incompressible Navier-Stokes equation [J]. Journal of Statistical Physics, 1997, 88(3-4):927-944.
- [14] CHEW Y T, SHU C, PENG Y. On implementation of boundary conditions in the application of finite volume lattice Boltzmann method
 [J]. Journal of Statistical Physics, 2002, 107(1-2);539-556.
- [15] SUCCI S. Lattice Boltzmann equation for fluid dynamics and beyond [M]. Oxford:Clarendon Press, 2001.
- [16] NOGUEIRA X, COLOMINAS I, CUETO-FELGUEROSO L, et al. Resolution of computational aeroacoustics problems on unstructured grids with a higher-order finite volume scheme [J]. Journal of Computational and Applied Mathematics, 2010, 234(7):2089-2097.