

复杂多相流动分子动力学模拟在 GPU 上的实现

陈飞国^{①②}, 葛蔚^{①*}, 李静海^①

① 中国科学院过程工程研究所多相复杂系统国家重点实验室, 北京 100190;

② 中国科学院研究生院, 北京 100049

* 联系人, E-mail: wge@home.ipe.ac.cn

收稿日期: 2008-06-12; 接受日期: 2008-10-08

国家自然科学基金(批准号: 20336040, 20221603, 20490201)和中国科学院知识创新工程(批准号: KGCXZ-YW-124)资助项目

摘要 利用 CUDA(Compute Unified Device Architecture)技术进行了图形处理器(GPU)上的分子动力学(MD)模拟. 在一片 Tesla C870 上, 其速度 20~60 倍于 Intel Xeon 5430 CPU 之单核, 最高可达 150 Gflops. 通过方腔流及颗粒-气泡接触等实例初步展示了此方式从微观上模拟介观行为的能力.

关键词

多相流动
分子动力学
CUDA
GPU
并行计算

1 前言

分子动力学(molecular dynamics, MD)模拟是随着计算机技术的发展而兴起的一种科学计算方法, 从 1956 年 Alder^[1]首次使用以来, 现已应用到广泛的领域中, 如医药、材料、能源、机电等. 随着纳米、微机电和微化工等技术的兴起, 纳微流动的 MD 模拟近年来也成为热点. 传统连续流体力学难以处理和解释这些尺度上的独特性质与现象, 而流动归根结底是流体分子的集体行为, MD 模拟能详细跟踪每个分子的运动, 并通过分析速度、温度等统计性质阐释理论中的难点、发现新的机理, 故日益受到重视. 但计算能力一直是制约此研究发展的瓶颈.

现在这方面很多成熟算法都基于传统体系结构的中央处理器(CPU), 但其发展已显颓势: “摩尔定律”^[2]难以为继, 而应用对计算能力的需求增长则远高于“摩尔定律”. 而 90 年代发展起来的图形处理器(GPU)的计算能力现已远高于 CPU 一到两个量级, 并且描述 GPU 速度发展的“黄氏定律”(以 NVIDIA 公司总裁黄仁勋命名)也比“摩尔定律”要快, 并具有性价

比和计算密度高、低能耗等优势. 如何在 MD 流动模拟中利用 GPU 的强大能力已成为一个重要的现实课题.

从硬件结构上分析, 大量能够并行运算且共享(较小的)多级高速显存的线程处理器是 GPU 快速计算的依靠. 它适合许多类似图形操作的计算和数据密集型应用, 其典型正是粒子信息简单、运动相对独立、作用可叠加和近程相关而计算量大、时间长的 MD 模拟. 但传统 GPU 对非图形应用支持有限, 只能通过图形 API 编程, 内存带宽较低且访问限制多, 从而制约了其性能发挥. CUDA 的发布^[3]为此提供了新的软硬件架构. CUDA 把 GPU 直接视作数据并行计算设备而不再将计算映射到图形操作上, 并以类 C 语言的方式给开发者更大自由来实现 GPU 算法. CUDA 还能结合 OpenMP, MPI 和 PVM 等其他并行方式在节点内和节点间继续扩展计算能力.

目前, 已有一些基于 CUDA 的 GPU 计算实例, 如天体物理中的 N-body 模拟^[4], LBM^[5], MD^[6,7]等, 发挥的 GPU 性能达到了 CPU 单核性能的几十到上百倍

之间, 但完全在 GPU 上进行、并且采用大量 GPU 并行或针对多相系统的 MD 模拟尚未见报道. 本文将报告基于 CUDA 在 GPU 上进行的多相纳微流动的并行 MD 模拟. 为此, 第 2 节介绍 MD 模拟及其 GPU 实现的具体过程. 第 3 节测试该实现的正确性并比较主流 GPU 和 CPU 发挥的性能. 第 4 节初步应用 GPU 实现了方腔流和微粒在液流中与气泡作用的 MD 模拟. 第 5 节总结全文并展望 GPU 通用计算的前景.

2 方法

2.1 分子动力学(MD)模拟

在本文模拟的多相体系中, 各相内及相间的分子间作用均采用截断的 Lennard-Jones 势, 即

$$\phi(r_{ij}) = \begin{cases} 4\epsilon \left[a \left(\frac{\sigma}{r_{ij}} \right)^{12} - b \left(\frac{\sigma}{r_{ij}} \right)^6 \right], & r_{ij} = |\mathbf{r}_i - \mathbf{r}_j| \leq r_c, \\ 0 & r_{ij} > r_c, \end{cases} \quad (1)$$

其中, ϵ 为作用势能, σ 为作用特征长度(一般为分子直径), r_{ij} 为分子 i 到分子 j 的距离; 系数 a 、 b 表示斥力项和引力项的相对大小, 一般情况 $a = b = 1$, 而 $a = 1$ 和 $b = 0$ 表示只有斥力作用. 按常规, 本文取 $\sigma = 1$, $\epsilon = 1$ 和分子质量 $m = 1$. 作为简化模型, 固体分子还受一种维持其晶格结构的“弹簧”力^[8]

$$\phi(s) = 1/2 Cs^2, \quad (2)$$

其中 C 为倔强系数, s 为分子在晶格上振动时离开其平衡位置的距离.

针对 GPU 计算特点, 本文用元胞列表法来高效组织和检索模拟数据. 模拟区域被分成若干个大小等于或稍大于截断距离 r_c 的网格. 这样各分子仅与相同或相邻网格内的分子作用, 此网格数与体系大小无关, 三维时共 27 个、二维时共 9 个, 故本算法的计算量与分子数 N 成正比, 即 $O(N)$.

在实现元胞列表法的各种方案中, 用链表储存网格内分子的信息可以在不用修改参数的情况下适应任意分子数密度, 但这种方式对内存的访问是随机的而且难以在同一网格内并行. 另一种用预先给定大小的数组储存网格内分子信息的方法可避免链表的缺陷、连续和并行地访问内存, 但这些数组能

容纳网格中可能的最大分子数, 占用内存较大. 本文采用后一种实现方案, 使 GPU 计算中每个线程块(block)对应一个网格, 而每个线程(thread)对应其中一个分子.

2.2 GPU-MD 算法

整个元胞列表法 MD 模拟可分解成分子运动、网格映射更新、计算分子间作用力等几个主要部分. 分子运动的路径积分采用蛙跳格式^[9], 即

$$\begin{aligned} \mathbf{v}(t + \Delta t/2) &= \mathbf{v}(t) + \frac{\mathbf{f}(t)}{2m} \Delta t, \\ \mathbf{r}(t + \Delta t) &= \mathbf{r}(t) + \Delta t \mathbf{v}(t + \Delta t/2), \\ \mathbf{v}(t + \Delta t) &= \mathbf{v}(t + \Delta t/2) + \frac{\mathbf{f}(t + \Delta t)}{2m} \Delta t, \end{aligned} \quad (3)$$

该积分在时驱 MD 中需分两步完成: (1) 将分子 t 时的速度 $\mathbf{v}(t)$ 根据当时的受力 $\mathbf{f}(t)$ 更新到 $\mathbf{v}(t + \Delta t/2)$, 从而将其位置 $\mathbf{r}(t)$ 更新到 $\mathbf{r}(t + \Delta t)$. (2) 按 $t + \Delta t$ 时的各分子位置计算受力 $\mathbf{f}(t + \Delta t)$, 把速度由 $\mathbf{v}(t + \Delta t/2)$ 更新到 $\mathbf{v}(t + \Delta t)$, 完成分子位置、速度同步. 算法流程如图 1 所示.

我们尝试了只用 GPU 处理最耗时的分子间作用, 但因主存和显存间频繁的数据传输成为性能瓶颈(PCI-Ex16 总线带宽为 4 GB/s 远小于显卡内部的 76.8 GB/s¹⁾), 效果并不理想. 本文用 GPU 处理上述 MD 算法中的所有模块, 这样只需 CPU(host thread)在开始计算前读入模拟参数和分子信息、映射到各相应网格内并按此复制到显存, 供 GPU 计算时调用.

分子运动在 GPU 中的计算较简单, 线程块内的每个线程更新对应的分子位置和速度信息即可, 但因分子位置变化, 需重新映射到网格. 为此, 首先计算网格内的各分子当前归属网格编号, 这一步可以各线程并行完成. 随后将原来的网格映射数组内不属于本网格的分子移出, 再遍历邻居网格(不包括本网格, 三维时共 26 个), 将属于本网格的分子插入到映射数组中; 为了避免内存写入冲突, 这两个步骤在每个线程块中只由一个线程操作. 在移出不属于本网格的分子前需要备份映射数组, 以确保在更新完成前映射数组是完整的, 在下一步插入新分子时可以找到待插分子的信息.

1) Tesla C870 显存带宽, 引自 http://www.nvidia.com/object/tesla_c870.html

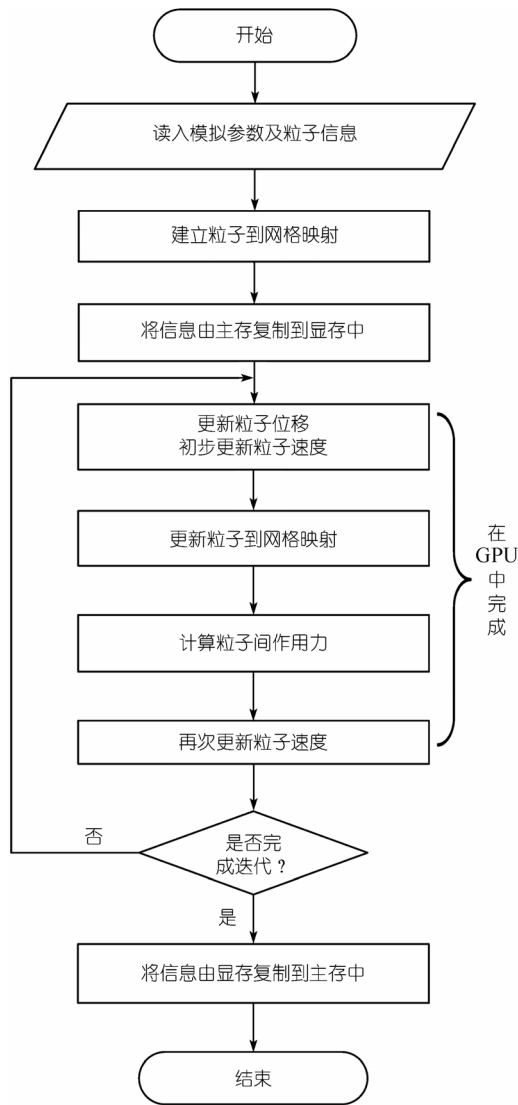


图 1 分子动力学模拟 GPU 计算流程图

MD 模拟中, 分子间力的计算最耗时. 线程 i 将网格 c_0 和邻居网格 c_n 的分子映射数组的第 i 号分子由全局内存读到共享内存中. 在同一线程块的各线程均完成数据加载后(通过线程块内的同步确保完成), 线程 i 计算 c_0 的分子 i 与 c_n 内所有分子的作用. 如此完成整个线程块 c_0 和 c_n 内所有分子间作用的计算. 三维时 $n = 0, 1, \dots, 26$, 即 c_0 有 27 个邻居网格(包括 c_0 本身). 最后, 将更新后的共享内存中的分子信息写回各自对应的全局内存.

2.3 GPU-MD + MPI 并行计算方法

虽然单个 GPU 的性能已非常强劲, 但对大规模 MD 模拟还远远不够. 本文以扩展性较好的区域分解和常用的消息传递接口(Message Passing Interface, MPI)协议实现各节点间 GPU 的并行计算. 每步内各进程只计算所分配区域内所有分子的受力、速度和位移, 当分子运动到新的区域时就分配给新的处理器计算. 为了计算其内分子的受力, 处理器只需邻近处理器中分子的信息, 如图 2.

在我们的算法里, CUDA 和 MPI 间没有关联和制约, 是相互透明的, 算法实现也是独立的. 进程内的所有计算均在 GPU 上完成, 而进程间的通信通过 CPU 完成, 即为 GPU-CPU-CPU-GPU 的模式. 由于 GPU 的速度极快, 这里的性能瓶颈在于 GPU 到 CPU 的带宽(PCI-Ex16 总线)及节点间的网络带宽(本文采用千兆以太网). 为此需要让 GPU 计算尽可能长的时间(让一个进程容纳尽可能多的分子, 即粗粒度并行)或使计算与通信叠加, 以减少通信在整个程序运行

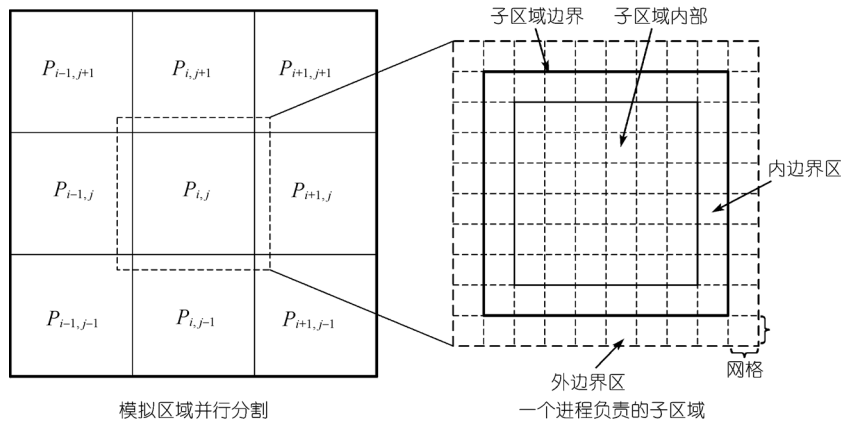


图 2 (二维)区域分解示意图

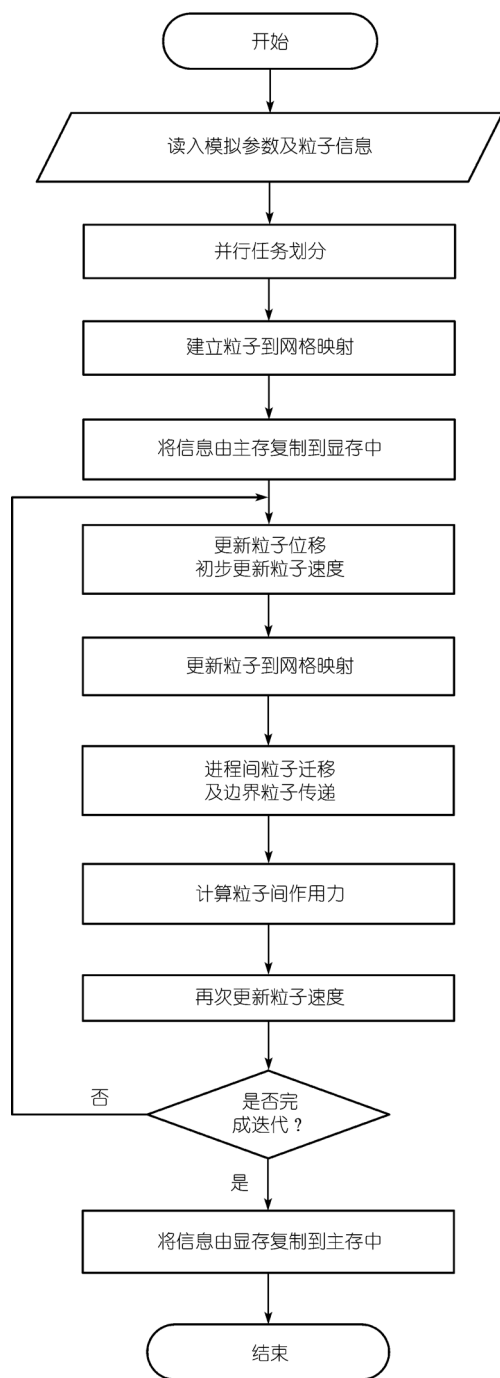


图 3 分子动力学模拟并行 GPU 计算流程图

中的时间比例. 并行 MD 模拟的流程如图 3.

3 GPU 性能的发挥

虽然 GPU 提供的理论计算峰值很高, 但实际发

挥的性能很大程度取决于程序结构. 数据读写模式对 GPU 计算性能发挥有很大影响. 在 CUDA 中, GPU 计算所需数据存储方式分片上 (on-chip) 和片下 (off-chip). 对 Tesla C870, 片下内存包括全局内存 (1.5 GB, 可读写, 速度慢), 常量内存 (64 KB, 只读, cached) 和纹理内存 (最大可映射 2^{27} 线性内存, 只读, cached); 片上内存也可称作 (处理器) 当地内存, 每块多处理器 (multiprocessor) 包含 8192 个 32 位寄存器和 16KB 共享内存 (若没有访问冲突, 速度与寄存器相当), 还有一些指令存储空间等 (cached, 对用户不可见). GPU 访问全局内存的带宽很大 (76.8 GB/s), 但延迟较大 (400~600 个时钟周期), CUDA 提供一种合并访问全局内存来掩盖延迟的方法, 可以把访问效率提高一个量级以上^[3].

尽量使用当地内存 (共享内存和寄存器) 及合并访问全局内存能让 GPU 发挥计算密集优势, 提高程序效率. 在我们的 GPU-MD 程序中, 一个线程块对应一个网格, 一个线程对应一个分子. 一个分子从全局内存读到共享内存后, 跟它的邻居分子间的作用力全部计算完后写回全局内存, 同时计算邻居分子与同个线程块内所有分子间的作用. 计算分子与它的邻居 (一般来说邻居数为数百个) 作用过程中, 分子的信息一直保留在共享内存中, 很多次计算才对应一次读写全局内存操作, 而且在相同线程块内的各个线程读到的全局内存空间是连续的, 这样处理可以极大地改善数据访问效率并把 GPU 计算性能发挥出来.

在 GPU-MPI 并行模式下, 在计算进程边界处的分子受力时, 要从邻居进程获得所需的分子信息, 而分子信息每步都会改变, 因此进程一般每步都需要与相邻进程通信, 具有局部频繁通信而传输量不大的特点. 单相 MD 模拟的负载较均衡, 同步时间短, 并行效率较高. 为形象比较 GPU 和 CPU 的性能, 我们用 CPU 和 GPU 分别模拟计算了一个简单分子体系的平衡状态动力学过程. 具体模拟参数列于表 1 中.

在上述 MD 模拟中, Tesla C870 约能发挥 45 Gflops 的有效计算性能 (指计算分子对相互作用时的浮点操作数), 而相同算法下, Intel Xeon 5430 CPU 单核能发挥 2.0 Gflops 左右, GPU 约为 CPU 的 20~30 倍. 若考察极端情况, 即只测试耗时最长的分子间作用

表 1 性能比较采用的模拟参数

	势能参数					体系参数				
	r_c	ε	σ	a	b	模拟区域	温度	数密度	分子数	时间步长
简单 MD	5	1	1	1	1	100×100×100	1	0.729	7.29×10^5	0.005
只算力 MD	5	1	1	1	1	100×100×100	1	1	1×10^6	0.005

力计算, GPU 更占优势, 约能发挥 150 Gflops, 占其可利用计算峰值(346 Gflops)的 40%多, 而 CPU 能发挥 2.4 Gflops, GPU 的计算能力达到 CPU 的 60 倍之多! 我们在 100 个节点(共 200 块 C870 卡)上并行发起 200 个进程, 仅计算力的性能可达近 30 Tflops, 而简单 MD 模拟发挥的性能为 8~12 Tflops, 实测通信时间约为总时间的 10%~25%(依赖于算例和并行划分粒度), 具体结果比较列于表 2.

需要指出, 无论 GPU 还是 CPU, 发挥的性能与 MD 算例的特性也有很大关系. 上述对比的 CPU 程序利用 SSE 指令优化过, 就分子间作用计算而言, 优化后的性能可提高 1~2 倍; 但对于 MD 计算, 由于在计算中存在条件判断指令, 只能提高 30%左右. 在我们的 GPU 算法里, 只要线程块数相同, 计算时间就相同, 也就是说相同分子数, 数密度越高, 每个网格内分子数越多, 相应的线程块数越少, 计算时间越少. 在上述算例中, 截断距离 $r_c = 5$, 一个网格内平均含有约 100 个分子, 只有少量的线程处理不相互作用的分子, 计算量浪费已不多. 如当 r_c 增大到 20 时(网格大小没变化, 而通过搜索周边更多邻居网格来实现), CPU 的性能由于数据读写频繁反而出现些许下降; 但 GPU 的性能却能提高 1.6 倍, 计算力的时间比例也由 67.6%提高到 98.7%. 这说明 GPU 很适合像多体问

题那样需要精确计算长程力的问题.

4 应用 GPU-MD 算法

4.1 方腔流模拟

方腔流是流体力学的一个经典问题. 它形式简单, 但可形成复杂的流动现象, 如在封闭体内可出现角涡, 纵涡, 多态性, 转捩, 湍流等各种流体力学现象^[10]. 在驱动方腔内流体运动的运动顶板角上, 无滑移边界条件不适用, 形成奇点问题. 方腔流的这些特性吸引了很多研究者用各种方法进行研究, 近年来也有很多利用计算机模拟方腔流的工作, 如 CFD^[11], LBM^[12,13]及 MD^[14,15]等. MD 模拟方腔流无需人为假设边界条件, 其结果有助于认识方腔流中各种现象的机理.

我们将 GPU 应用于方腔流的 MD 模拟, 使得模拟体系的规模有较大提高, 在一定程度上达到了(亚)微米尺度, 沟通了连续的流体力学和离散的分子动力学. 具体模拟参数如表 3 所示. 模拟区域内共有 3.2×10^6 个粒子, 时间步长 $\Delta t = 0.005\tau$ (τ 为特征时间, $\tau = (m\sigma^2/\varepsilon)^{1/2}$), 该算例中由 4 个 GPU 并行计算, 各方向均采用 Maxwell 热边壁条件, 边壁温度保持在 $k_B T = 0.8\varepsilon$, 顶板速度 U 沿+X 方向.

从图 4 中可以明显看到, 在各截面均存在方腔流

表 2 MD 模拟运行时间和性能比较, 每个算例(或 MPI 并行的一个进程)模拟区域为 100×100×100

	运动时间 s/timestep	网格更新 s/timestep	力计算时间 s/timestep	通信时间 s/timestep	总时间 s/timestep	性能折算 Gflops	加速比
CPU	0.036	0.046	14.56		14.67	2.05	1
GPU	0.042	0.146	0.46		0.68	44.30	21.6
GPU+MPI	0.037	0.206	0.45	0.115	0.81	37.19	18.1
CPU, $r_c=20$	0.074	0.106	1054.83		1055.09	1.82	1
GPU, $r_c=20$	0.042	0.144	16.57		16.79	114.83	63.1
CPU,只算力			29.07		29.07	2.44	1
GPU,只算力			0.48		0.48	148.66	60.9

表 3 方腔流模拟参数

	密度	温度	黏度	声速	模拟区域大小	顶板速度	Reynold 数	Mach 数
	ρ	T	μ	c	(长×高×厚)	U	$R_e=U\rho H/\mu$	$Ma=U/c$
无量纲值	0.8	0.8	2.28	5.42	200×200×100	2	140.4	0.37
实际值 ^{a)}	1350 kg/m ³	96 K	2.08×10 ⁻³ P	861 m/s	68 nm×68 nm×34 nm	318 m/s	140.4	0.37

a)按氩(Argon)原子参数计算: $\sigma = 3.40 \times 10^{-10}$ m, $\epsilon = 1.67 \times 10^{-21}$ J, $m = 6.63 \times 10^{-26}$ kg, $\tau = 2.14 \times 10^{-12}$ s

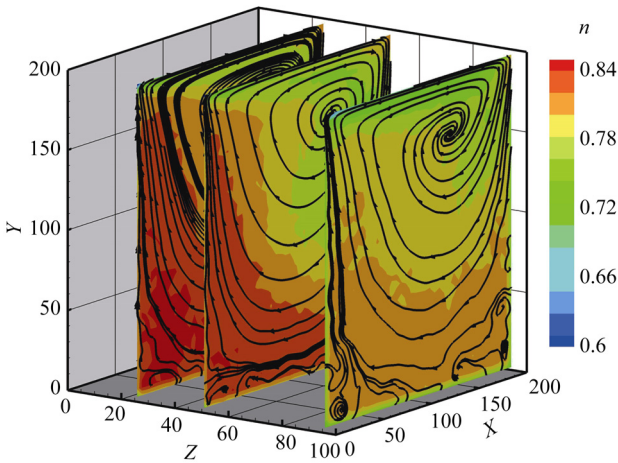


图 4 Z = 25, 50 和 95 处的剖面图
云图表示分子数密度分布, 线条表示流线

特有的主涡, 位置基本一致, 但形状不同. 实际上它在Z方向是一体的, 具有图5所示的三维结构^[10]. 图4

中还能看到底板角落附近的两个角涡. X = 100 和 Y = 100 剖面上的速度分布情况如图 6 所示, 长度为该位置速度大小而箭头是其方向, 其中驱动顶板在 Y = 200 位置. 上述结果与已有的实验和模拟报道定性相符, 但定量对比尚待更大规模和更长时间的模拟及相应的统计与分析.

4.2 多相流动

相对于单相, 多相纳微流动研究对纳米、材料、生物、微机电与微化工系统等技术领域更有实用性, 但也更有挑战性, 目前还没有成熟的理论方法, 而其 MD 模拟需要更多的计算资源且实现更加复杂. 因此, GPU 在这里的应用显得更有价值.

气液固三相体系中分子间两两作用的方式共有六种, 基于第 2 节的模型, 我们采用的具体作用参数为: 液体-液体间, $r_{cl}=5$, $\sigma_l=1$, $\epsilon_l=1$, $m_l=1$; 气体-气体间, 只有斥力项的 LJ 势, $r_{cg}=2$, $\sigma_g=0.5$, $\epsilon_g=1$, $m_g=1$; 固体-固体间作用势与液体相同, 固定“弹簧”倔强系数

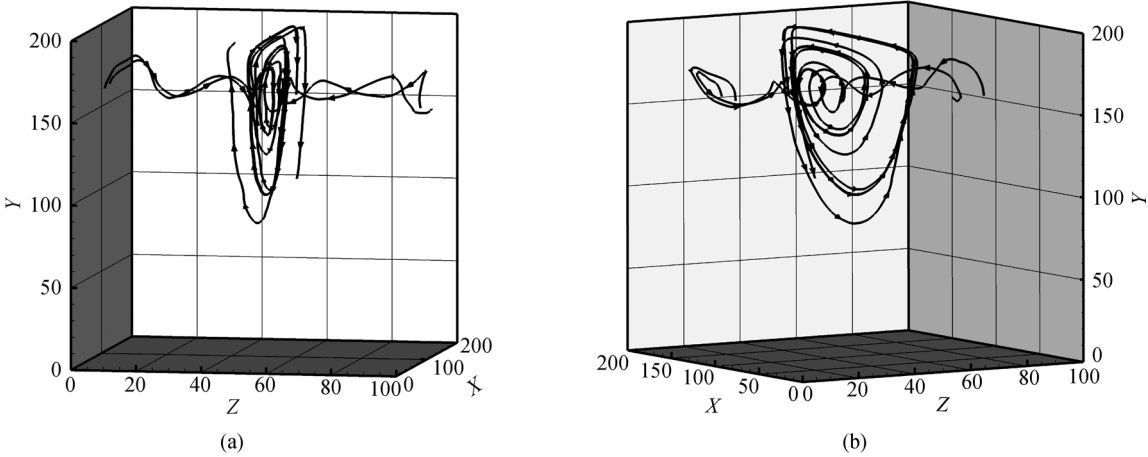
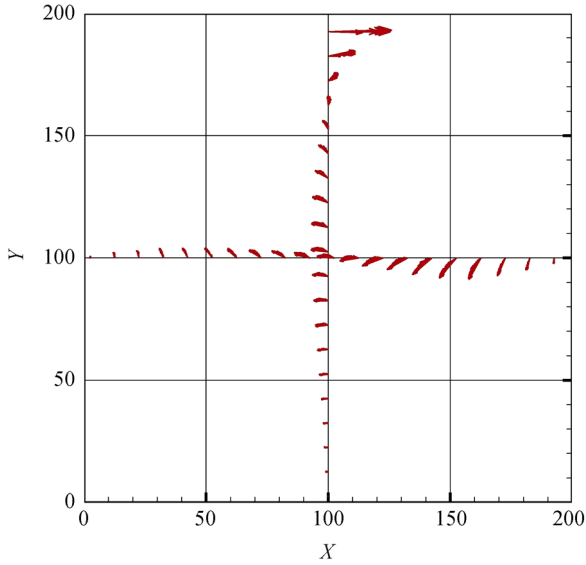


图 5 主涡的三维结构示意

(a)和(b)分别是两个角度的视图. 图中线条表示流场中主涡附近的两条流线, 可以看到在 Z 方向这两条流线贯穿将各位置主涡连成一体, 呈现明显的三维结构

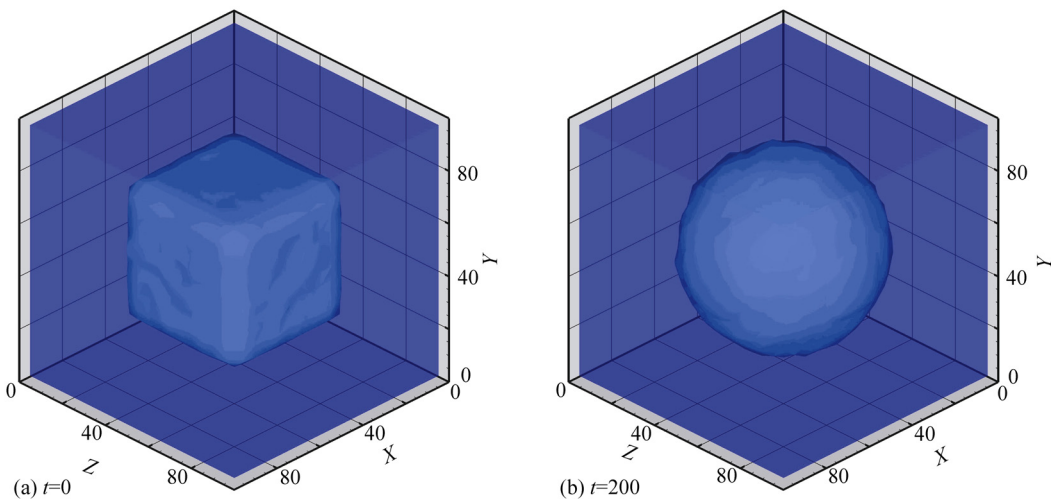

 图 6 $X=100$ 和 $Y=100$ 剖面速度矢量分布图

$C = 75$, 固液间的 LJ 势参数与液体或固体相同; 气液和气固间采用只有 12 次斥力项的 LJ 势, $r_c = 3$, $\sigma = 1$, $\epsilon = 1$.

为了检验所建立的三相计算模型的可行性, 我们首先模拟了气液两相中, 气泡由初始方形演化为平衡态下球形的过程. 纳米气泡已由 Gogotsi 等人^[16]在实验中观测到, 并且也有不少研究者用 MD 方法 (在 CPU 中计算) 模拟过纳米气泡的行为, 其中 Sushko 和 Cieplak^[17] 及 Chen 等人^[18] 直接模拟了气体分

子, 其他一些研究者^[19~22] 将气泡处理为空穴而没有直接模拟气体分子. 在我们的实例中, 模拟区域大小为 $100 \times 100 \times 100$, X , Y , Z 方向均采用周期边界条件. 气泡初始方形设置 (图 7(a)) 在区域中心, 大小为 $50 \times 50 \times 50$, 气体分子总数 $N_g = 91125$, 数密度 $n_g = 0.729$. 气泡周围的液体分子总数 $N_l = 625177$, 数密度 $n_l = 0.714$. 体系初始温度 $k_B T = 1$, 模拟时间步长 $\Delta t = 0.005$. 弛豫至 $t = 200$ 时, 方形气泡在界面张力作用下变成 (近似的) 球形 (图 7(b)), 此时气液界面能最小. 平衡状态下, 气泡直径 $D_b = 70.1$ (体积 $V_b = 1.80 \times 10^5$), 气体分子数密度 $n_g = 0.506$, 而对应的液体分子数密度 $n_l = 0.763$. 这里气液两相密度相差不大, 与常规状态不尽相同. 但实际的纳米气泡也会因为受到很大界面附加压力被严重压缩具有较高密度. 而“气泡”之所以认为是气态是考虑到构成气泡的粒子相互之间只有斥力没有吸引力. 当然由于简单的 L-J 型的势函数还难以表达复杂的界面作用, 本算例中气泡的刚性比实际纳米气泡要弱得多. 但从算法的角度看, 这个算例说明了多相分子动力学模拟在 GPU 上并行可行性.

在此成功的基础上, 我们尝试模拟气泡在液体槽流中与固定颗粒的作用. 具体设置如下: 模拟区域为 $80 \times 160 \times 80$ (两边是厚度为 3 的壁面, 由 76800 个数密度 $n_s = 1$ 的固体分子组成). 直径 $D_b = 50$ 的气泡 (含 45965 个气体分子) 设置在底端, 直径 $D_s = 20$ 的固体颗


 图 7 气泡由初始设置的方形 ($t=0$) 演变为平衡状态的 (近似) 球形 ($t=200$)

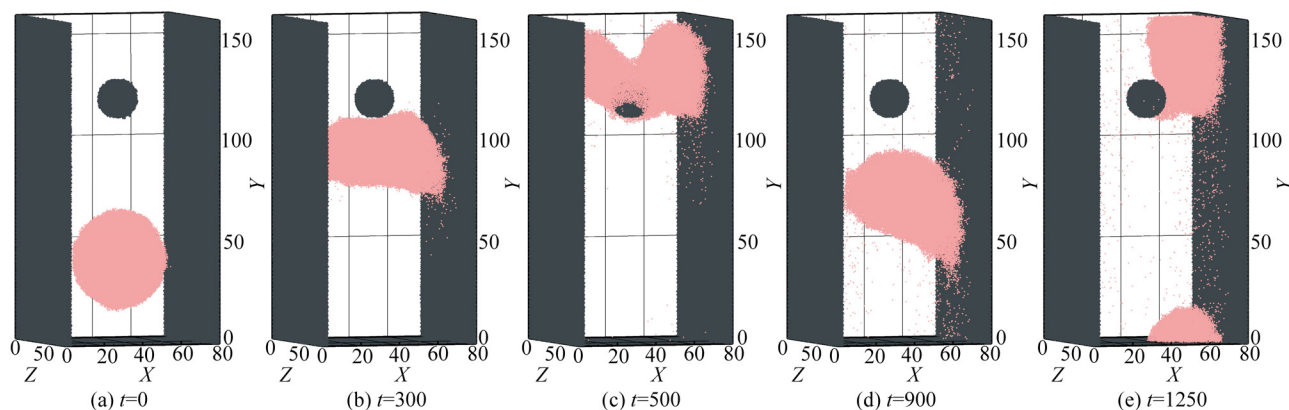


图 8 气液固三相模拟各时刻的快照

黑色表示气体分子, 灰色表示固体颗粒和壁面, 填充其间的液体未表示

粒(含 3544 个固体分子)固定在上部, 其他地方填充数密度 $n_l = 0.733$ 的液体分子, 总数 $N_l = 643240$. 体系初始温度设定为 $k_B T = 1$, 质量力 $g = 2 \times 10^{-3}$, 温控采用壁面分子速度标定方法, Y, Z 方向为周期性边界.

气体和液体在恒定质量力 g 作用下开始流动. 在气泡未接触上方的固定颗粒时, 受到周围液体的作用左右基本平衡, 故气泡沿中线运动, 并从初始的球形变成泡状. 如图 8 所示, 当气泡第一次接触固体颗粒时($t = 300$), 固体颗粒穿过气泡($t = 500$); 由于固体颗粒周围流场非对称, 气泡受力偏倚, 之后沿一侧随液体流动($t = 900$). 再次与固体颗粒接触时, 由于气液界面张力的限制而未被固体颗粒穿透($t = 1250$), 而后气泡贴在壁面运动. 此现象与人们设想的矿物浮选等过程中气固接触的一些典型过程相似^[23,24], 但还待进一步分析.

上述模拟只是用来说明本文建立的 GPU-MD 算法的应用. 模拟得到的现象也只是文中用到的作用势函数及参数下的数值结果, 这些势函数可能与真实情况不太相符, 因此无法与实际体系相比较. 但可

以作为一种理想的参考模型来考察纳微尺度下的多相流动. 另一方面, 借助这样的 MD 模拟可以实现某些物理实验难以进行的极端情况, 比如, 上例中的 g 约 10^{10} 倍于地球表面的重力加速度.

5 结论

现在的图形处理器(GPU)计算能力远超主流中央处理器(CPU), 通用 GPU 计算以高性价比的优势越来越受人们关注. 利用 CUDA 技术和 MPI 协议, 我们实现了多相纳微流动在 GPU 上的并行 MD 模拟, 获得了合理的结果. 计算表明, 目前主流的 GPU 可发挥 20~60 倍于主流 CPU 的计算能力, 并且粗粒度并行(CPU 间)和细粒度并行(GPU 内)的混合模式能获得近似线性的规模加速比, 是 MD 模拟中非常值得关注的新方式.

目前我们只考虑了分子间的 LJ 势, 而实际应用中的库仑力和化学键等将使算法更加复杂, 需要更深入的优化, 但总体来看, GPU 在 MD 模拟上的应用还有巨大的潜力.

致谢 在本工作进行期间, 作者得到了 NVIDIA 中国有限公司相关人士的大力帮助, 特此致谢.

参考文献

- 1 Alder B J, Wainwright T. Molecular dynamics by electronic computers. In: Prigogine I, eds. Proceeding of International Symposium on Transport Processes in Statistical Mechanics. Brussels: Interscience, New York: Wiley, 1956. 97—131

- 2 Moore G E. Craming more components onto integrated circuits. *Electronics*, 1965, 38(8): 114—117
- 3 NVIDIA. NVIDIA CUDA Compute Unified Device Architecture Programming Guide Version 2.0. 2008
- 4 Belleman R G, Bédorf J, Zwart S F P. High performance direct gravitational N-body simulations on graphics processing units II : An implementation in CUDA. *New Astron*, 2008, 13(2): 103—112 [\[DOI\]](#)
- 5 Tölke J. Implementation of a Lattice Boltzmann kernel using the compute unified device architecture developed by nVIDIA. *Comput Visual Sci*, 2008, DOI:10.1007/s00791-008-0120-2
- 6 van Meel J A, Arnold A, Frenkel D, Zwart S F P, Belleman R G. Harvesting graphics power for MD simulations. *Mol Simulat*, 2008, 34(3): 259—266 [\[DOI\]](#)
- 7 Anderson J A, Lorenz C D, Travesset A. General purpose molecular dynamics simulations fully implemented on graphics processing units. *J Comput Phys*, 2008, 227(10): 5342—5359 [\[DOI\]](#)
- 8 Liem S Y, Brown D, Clarke J H R. Investigation of the homogeneous-shear nonequilibrium-molecular-dynamics method. *Phys Rev A*, 1992, 45(6): 3706—3713 [\[DOI\]](#)
- 9 Frenkel D, Smit B. *Understanding Molecular Simulation: From Algorithms to Applications*. Orlando: Academic Press, 1996. 443
- 10 Shankar P N, Deshpande M D. Fluid mechanics in the driven cavity. *Annu Rev Fluid Mech*, 2000, 32(1): 93—136 [\[DOI\]](#)
- 11 Albensoeder S, Kuhlmann H C. Accurate three-dimensional lid-driven cavity flow. *J Comput Phys*, 2005, 206(2): 536—558 [\[DOI\]](#)
- 12 Hou S, Zou Q, Chen S, Doolen G, Cogley A C. Simulation of cavity flow by the Lattice Boltzmann method. *J Comput Phys*, 1995, 118(2): 329—347 [\[DOI\]](#)
- 13 Nie X, Robbins M O, Chen S. Resolving singular forces in cavity flow: Multiscale modeling from atomic to millimeter scales. *Phys Rev Lett*, 2006, 96(13): 134501—4 [\[DOI\]](#)
- 14 Qian T, Wang X-P. Driven cavity flow: from molecular dynamics to continuum hydrodynamics. *Multiscale Model Simul*, 2005, 3(4): 749—763 [\[DOI\]](#)
- 15 Koplik J, Banavar J R. Corner flow in the sliding plate problem. *Phys Fluids*, 1995, 7(12): 3118—3125 [\[DOI\]](#)
- 16 Gogotsi Y, Libera J A, Guvenç-Yazicioglu A, Megaridis C M. *In situ* multiphase fluid experiments in hydrothermal carbon nanotubes. *Appl Phys Lett*, 2001, 79(7): 1021—1023 [\[DOI\]](#)
- 17 Sushko N, Cieplak M. Motion of grains, droplets, and bubbles in fluid-filled nanopores. *Phys Rev E*, 2001, 64(21): 021061—8
- 18 Chen F, Ge W, Wang L, Li J. Numerical study on gas-liquid nano-flows with pseudo-particle modeling and soft-particle molecular dynamics simulation. *Microfluid Nanofluid*, 2008, 5(5): 639—653
- 19 Matsumoto M, Matsuura T. Molecular dynamics simulation of a rising bubble. *Mol Simulat*, 2004, 30(13-15): 853—859 [\[DOI\]](#)
- 20 Matsumoto M, Tanaka K. Nano bubble—Size dependence of surface tension and inside pressure. *Fluid Dynam Res*, 2008, 40(7-8): 546—553 [\[DOI\]](#)
- 21 Nagayama G, Tsuruta T, Cheng P. Molecular dynamics simulation on bubble formation in a nanochannel. *Int J Heat Mass Tran*, 2006, 49(23-24): 4437—4443 [\[DOI\]](#)
- 22 Park S H, Weng J G, Tien C L. A molecular dynamics study on surface tension of microbubbles. *Int J Heat Mass Tran*, 2001, 44(10): 1849—1856 [\[DOI\]](#)
- 23 Nguyen A V, Evans G M. Attachment interaction between air bubbles and particles in froth flotation. *Exp Therm Fluid Sci*, 2004, 28(5): 381—385 [\[DOI\]](#)
- 24 Omota F, Dimian A C, Blik A. Adhesion of solid particles to gas bubbles. Part 2: Experimental. *Chem Eng Sci*, 2006, 61(2): 835—844 [\[DOI\]](#)