

基于改进FPM法高维复杂多相分离过程的GPU并行计算研究

甄玉洁, 胥康, 蒋涛, 任金莲

GPU Parallelization Computation of High-Dimensional Multi-Phase Separation in Complex Domains Based on the Corrected FPM

ZHEN Yujie, XU Kang, JIANG Tao, and REN Jinlian

在线阅读 View online: <https://doi.org/10.21656/1000-0887.430147>

您可能感兴趣的其他文章

Articles you may be interested in

分数阶Cahn–Hilliard方程的高效数值算法

An Efficient Numerical Algorithm for Fractional Cahn–Hilliard Equations

应用数学和力学. 2021, 42(8): 832–840 <https://doi.org/10.21656/1000-0887.420008>

基于CUDA–GPU架构的超二次曲面离散单元并行算法

A Parallel Algorithm for Super–Quadric Discrete Elements Based on the CUDA–GPU Architecture

应用数学和力学. 2019, 40(7): 751–767 <https://doi.org/10.21656/1000-0887.390267>

三维薄结构热传导问题分层二次方程的多水平方法

A Multi–Level Method for Hierarchical Quadratic Discretizations of Thin–Walled Structures in 3D Heat Conduction

应用数学和力学. 2018, 39(6): 700–713 <https://doi.org/10.21656/1000-0887.380035>

三维海洋水动力计算 σ 坐标转换存在的问题及改进方法

Problems and Improved Methods of 3D Ocean Hydrodynamic Calculation With the σ Coordinate Transformation

应用数学和力学. 2019, 40(8): 840–855 <https://doi.org/10.21656/1000-0887.390344>

基于三维CFD–DEM的多孔介质流场数值模拟

Numerical Simulation of Flow Fields in Porous Media Based on the 3D CFD–DEM

应用数学和力学. 2017, 38(10): 1093–1102 <https://doi.org/10.21656/1000-0887.370326>

功能梯度板中Griffith裂纹尖端应力场的三维解析研究

3D Analytical Solutions of Stress Fields at Griffith Crack Tips in Functionally Graded Plates

应用数学和力学. 2021, 42(1): 36–48 <https://doi.org/10.21656/1000-0887.410143>



关注微信公众号, 获得更多资讯信息

基于改进 FPM 法高维复杂多相分离过程的 GPU 并行计算研究*

甄玉洁, 胥康, 蒋涛, 任金莲

(扬州大学 数学科学学院, 江苏 扬州 225002)

(我刊编委赵景军推荐)

摘要: 为了高效、准确地模拟高维多元 Cahn-Hilliard (C-H) 方程描述的复杂相分离过程, 该文发展了一种基于纯无网格改进有限点集法 (corrected finite pointset method, CFPM) 和 CPU-GPU 异构的快速并行算法 (简称为 CFPM-GPU). CFPM-GPU 的构造过程为: ① 基于 Taylor 展开和加权最小二乘思想, 采用 Wendland 权函数推导出空间一/二阶导数的有限点集法 (finite pointset method, FPM) 格式; ② 将多元 C-H 方程中四阶导数分为两个二阶导数, 依次运用 FPM 对其离散得到 C-H 的改进 FPM 法 (CFPM); ③ 基于 CUDA 的单个 GPU 架构, 首次给出了 CFPM 的一种并行算法以提高计算效率. 数值研究中, 首先对二维径向或三维球对称 C-H 方程描述的相分离基准算例进行了求解, 并与可靠结果作对比验证了提出的并行算法的准确性和高效性, 单个 CPU-GPU 异构并行计算效率约是串行情况的 160 倍; 其次, 运用 CFPM-GPU 对复杂区域上二维/三维的两相或三相分离现象进行数值预测, 并与其他方法结果做比较. 数值结果表明, 给出的 CFPM-GPU 能准确、高效地模拟二维/三维复杂区域上的多相分离过程.

关键词: FPM 格式; 三维多相分离; 多元 Cahn-Hilliard; GPU 并行

中图分类号: O246 **文献标志码:** A **DOI:** 10.21656/1000-0887.430147

GPU Parallelization Computation of High-Dimensional Multi-Phase Separation in Complex Domains Based on the Corrected FPM

ZHEN Yujie, XU Kang, JIANG Tao, REN Jinlian

(School of Mathematical Sciences, Yangzhou University, Yangzhou, Jiangsu 225002, P.R.China)

(Recommended by ZHAO Jingjun, M. AMM Editorial Board)

Abstract: Based on the corrected finite pointset method (CFPM) with CPU-GPU heteroid parallelization (CFPM-GPU), a high-efficiency, accurate and fast parallel algorithm was developed for the high-dimensional phase separation phenomena governed by the multi-component Cahn-Hilliard (C-H) equation in complex domains. The proposed parallel algorithm with the CFPM-GPU was built in a process like: ① introduce the Wendland weight function into the discretization of the finite pointset method (FPM) scheme for the 1st/2nd spatial derivatives, based on the Taylor series and the weighted least square concept; ② use the above FPM scheme twice to approximate the 4th spatial derivative in the C-H equation, which is called

* 收稿日期: 2022-04-26; 修订日期: 2022-06-13

基金项目: 国家自然科学基金 (11501495); 中国博士后科学基金 (2015M581869; 2015T80589); 江苏省自然科学基金 (BK20150436); 国家科技支撑计划 (2015BAD24B02-02)

作者简介: 甄玉洁 (1996—), 女, 硕士生 (E-mail: 1056700190@qq.com);

蒋涛 (1978—), 男, 副教授, 博士 (通讯作者. E-mail: jtrjl_2007@126.com).

引用格式: 甄玉洁, 胥康, 蒋涛, 任金莲. 基于改进 FPM 法高维复杂多相分离过程的 GPU 并行计算研究[J]. 应用数学和力学, 2023, 44(1): 93-104.

the CFPM method; ③ for the first time establish an accelerating parallel algorithm for the CFPM with local matrices by means of a single GPU card based on the CUDA programming. Two benchmark problems of 2D radially and 3D spherically symmetric C-H equations were first solved to test the accuracy and high-efficiency of the proposed CFPM-GPU, and the acceleration ratio of the GPU parallelization to the single CPU computation is about 160. Subsequently, the proposed CFPM-GPU was used to predict the 2D/3D multi-phase separation phenomena in complex domains, and the prediction was compared with other numerical results. The numerical results show that, the proposed CFPM-GPU is valid and high-efficiency to simulate the 2D/3D multi-phase separation cases in complex domains.

Key words: FPM scheme; 3D multi-phase separation; multi-component Cahn-Hilliard; GPU parallelization

引言

含高阶导数的多元 Cahn-Hilliard (C-H) 方程常被用来描述多相流体力学或热力学等领域中的复杂非线性相分离现象^[1-4]. 近些年来, 多元 C-H 的高效性数值模拟方法研究受到许多国际学者的关注, 目前, 已有多种基于网格类方法被成功地用于单变量 C-H 方程下非线性扩散或高维多元 C-H 方程下相分离过程的数值预测, 如有限元法^[5]、有限差分法^[6-8]和多重网格法等^[9-10]. 上述方法对多元 C-H 描述的多维复杂非规则相界面的模拟实现比较复杂, 且不易实施多 CPU 或异构图形处理器 (GPU) 的并行计算. 于是, 近年来不完全依赖网格的纯无网格法在国际科学计算领域中受到普遍关注. 相较于网格类方法, 无网格方法^[11-15]具有易推广到复杂区域问题求解和高维问题并行计算实施的优点^[15-20], 如有限点集法 (finite pointset method, FPM)^[21-24]和光滑粒子流体力学 (简称 SPH) 法等^[15,20-23].

然而, 上述纯无网格 FPM 和 SPH 法推广应用到 C-H 方程下多相分离过程的模拟在国际上还鲜有报道. 笔者在 2020 年首次将 FPM 扩展到二维单分量 C-H 方程的求解, 并发展了基于局部加密的 LR-FPM 法 (见文献 [24]), 将其直接推广应用到多元 C-H 方程下三维复杂多相分离过程的模拟涉及巨大计算耗时问题, 且计算精度有待进一步提高. FPM 数值模拟多元 C-H 下三维多相分离过程较网格类方法的主要优点在于: ① 易处理高维复杂相界面和易离散求解四阶空间导数; ② 易并行计算实施.

基于上述分析, 拓展 FPM, 采用 Wendland 权函数, 引入基于 CUDA 的 CPU-GPU 异构并行计算, 提出了一种能够高效、准确地模拟多元 C-H 方程下高维复杂多相界面分离过程的改进 FPM GPU 并行算法 (CFPM-GPU). 首先采用 Wendland 权函数的 FPM 对四阶导数分裂成的两个二阶空间导数依次离散; 其次准确施加 Neumann 边界条件; 最后, 采用基于 CUDA 语言的单个 CPU-GPU 异构协同的并行计算方式给出上述离散过程的 GPU 并行算法以加速计算模拟. 本文并行计算模拟中采用一个型号为 TITAN RTX 的 GPU, 其包含 24 GB 显存和 4608 个 CUDA 核心数. 数值算例中首先展现了方法的准确性和并行计算效果; 其次对二维复杂二或三相分离过程, 以及三维复杂二相分离过程进行了模拟预测.

1 三元 C-H 方程

本文主要研究多元 C-H 描述的相分离现象^[1,10], 考虑如下三元 C-H 方程在区域 $\Omega \subset R^d$ (维数 $d=2$ 或 3) 内的形式:

$$\begin{cases} \frac{\partial \phi_i}{\partial t} = \Delta \mu_i, \\ \mu_i = f(\phi_i) - \varepsilon^2 \Delta \phi_i + \beta(\phi_1, \phi_2, \phi_3), \end{cases} \quad i = 1, 2, 3, \quad (1)$$

在边界 $\partial\Omega$ 上满足齐次 Neumann 边界条件

$$\mathbf{n} \cdot \nabla \phi_i = 0, \quad \mathbf{n} \cdot \nabla \mu_i = 0, \quad (2)$$

其中, $\phi_i(\mathbf{x}, t)$ ($i = 1, 2, 3$) 为三元混合物中每个分量的摩尔数, \mathbf{n} 为边界上的单位法向量. 根据质量守恒律, 摩尔数总和须满足

$$\phi_1(\mathbf{x}, t) + \phi_2(\mathbf{x}, t) + \phi_3(\mathbf{x}, t) = 1, \quad (3)$$

$f(\phi_i) = F'(\phi_i) = \phi_i(\phi_i - 0.5)(\phi_i - 1)$ 和 Lagrange 乘子 $\beta(\phi_1, \phi_2, \phi_3) = -\frac{1}{3} \sum_{i=1}^3 f(\phi_i) = -\phi_1\phi_2\phi_3$, 此处 $F(\phi_i) = 0.25\phi_i^2(\phi_i - 1)^2$ 满足 Helmholtz 自由能^[4] 泛函:

$$\xi(\phi_1, \phi_2, \phi_3) = \int_{\Omega} \sum_{i=1}^3 \left(F(\phi_i) + \frac{\varepsilon^2}{2} |\nabla \phi_i|^2 \right) dx. \quad (4)$$

本文中取梯度能量系数 $\varepsilon = 4d_0/[4\sqrt{2}\tanh^{-1}(0.9)]$ (d_0 为区域上离散点的初始间距).

在本文数值模拟中, 利用 ϕ_3 表示一个任意复杂区域 $\tilde{\Omega} \subset \Omega$ (相分离现象发生在区域 $\tilde{\Omega}$ 内), $\tilde{\Omega}$ 区域上 $\phi_3 \approx 0$, 其余区域 $\Omega \setminus \tilde{\Omega}$ 上 $\phi_3 \approx 1$, 在区域 $\tilde{\Omega}$ 边界上考虑 $\phi_3 \approx 0.5$ (可详见文献 [10]). 此外, 根据式 (3) 可得 $\phi_2 = 1 - \phi_1 - \phi_3$, 因此, 方程 (1) 描述的相分离过程可以简化为求解关于 ϕ_1 的 C-H 方程:

$$\begin{cases} \frac{\partial \phi_1}{\partial t} = \Delta \mu_1, \\ \mu_1 = f(\phi_1) - \varepsilon^2 \Delta \phi_1 - \phi_1(1 - \phi_1 - \phi_2)\phi_3. \end{cases} \quad (5)$$

2 CFPM-GPU 并行算法

本文主要考虑多元 C-H 方程下高维复杂多相界面分离过程的纯无网格法模拟, 基于 CUDA 的单个 GPU 并行计算, 结合 Wendland 权函数的优点^[17], 拓展文献 [24] 中二维 FPM 离散过程得到一种能够准确、高效模拟多元高维 C-H 方程的 CFPM-GPU 并行算法.

2.1 三维 CFPM 离散格式

对带 Neumann 边界条件 (2) 含四阶导数三维 C-H 方程 (5) 的求解, 拓展应用文献 [24] 里对二维情况下单个 C-H 方程连续应用两次 FPM 的离散过程, 时间更新采用二阶预估校正格式. 基于 Taylor 展开和加权最小二乘思想的 FPM 主要用来离散函数空间一阶/二阶导数, 其详细离散过程参见文献 [25-27]. 相对于网格类方法, 纯无网格 FPM 更易实施非均匀或局部加密节点分布计算的优点也已在文献 [24] 中对二维问题进行了详细分析. 文献 [24] 中 FPM 离散过程采用样条权函数, 而根据文献 [17] 知, Wendland 权函数较样条函数具有更好的光滑性和较高的数值精度, 本文的离散过程采用 Wendland 权函数.

以三维空间上均匀布点为例, 采用的 Wendland 权函数^[17] 形式如下:

$$w(\mathbf{x}_j - \mathbf{x}, h) = w_d \begin{cases} [2 - (r/h)]^4 (2(r/h) + 1), & 0 \leq r/h < 2, \\ 0, & r/h \geq 2, \end{cases} \quad (6)$$

其中, $r = |\mathbf{x}_j - \mathbf{x}|$ 为支持域半径, j 代表位置点 \mathbf{x} 处支持域内的相邻点, w_d 为权函数系数, 其在二维、三维空间中分别为 $7/(64\pi h^2)$ 和 $21/(2^{11}\pi h^3)$, h 为光滑长度, 在离散节点不动的情况下常取 $h \approx 0.95d_0$ (d_0 为离散点初始间距).

由文献 [24-27] 知 FPM 离散过程中, 对一个函数 $f(\mathbf{x})$ 在计算区域的任意分布离散点 \mathbf{x} 处的一阶/二阶导数的近似值, 通过 \mathbf{x} 点在其支持域内所有相邻节点 \mathbf{x}_j ($j = 1, 2, \dots, n$) (n 为相邻节点数) 上进行 Taylor 级数展开和加权最小二乘思想过程得到的一个线性方程组求解而获得, 即需求的局部线性方程组为

$$(\mathbf{M}^T \mathbf{W} \mathbf{M}) \mathbf{a} = (\mathbf{M}^T \mathbf{W}) \mathbf{b}, \quad (7)$$

其中, 上标符号 “T” 为转置, $\mathbf{M}^T \mathbf{W} \mathbf{M}$ 为 9×9 的局部系数矩阵; $\mathbf{a}_{9 \times 1}$ 为 \mathbf{x} 处函数一阶/二阶导数形成的未知矢量, $\mathbf{a} = (f_x, f_y, f_z, f_{xx}, f_{xy}, f_{xz}, f_{yy}, f_{yz}, f_{zz})^T$; $\mathbf{b}_{(n+1) \times 1}$ 为所有相邻节点处与 \mathbf{x} 处函数差和施加边界条件构成的已知矢量, $\mathbf{b} = (f_1 - f, f_2 - f, \dots, f_n - f, 0)^T$; 对角矩阵 $\mathbf{W}_{(n+1) \times (n+1)}$ 对角线上元素为 “ $w_1, w_2, \dots, w_n, 1$ ”; 矩阵 $\mathbf{M}_{(n+1) \times 9}$ 为

$$\mathbf{M}_{(n+1) \times 9} = \begin{pmatrix} dx_1 & dy_1 & dz_1 & 0.5dx_1^2 & dx_1dy_1 & dx_1dz_1 & 0.5dy_1^2 & dy_1dz_1 & 0.5dz_1^2 \\ dx_2 & dy_2 & dz_2 & 0.5dx_2^2 & dx_2dy_2 & dx_2dz_2 & 0.5dy_2^2 & dy_2dz_2 & 0.5dz_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ dx_n & dy_n & dz_n & 0.5dx_n^2 & dx_ndy_n & dx_ndz_n & 0.5dy_n^2 & dy_ndz_n & 0.5dz_n^2 \\ n_x & n_y & n_z & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

单位外法向量 $\mathbf{n} = (n_x, n_y, n_z)$.

本文数值模拟中函数 $f(\mathbf{x})$ 为方程 (5) 中函数 ϕ 或 μ . 值得注意的是, 上述线性方程组的求解过程需要求解两

次来得到四阶空间导数的近似值,再结合二阶精度的时间离散格式可得方程(5)中 ϕ_1 新时间层里的数值解,然后可得 ϕ_2 ,该过程称为高维多元 C-H 方程的 CFPM 离散格式。

2.2 单个GPU 并行加速算法

众所周知,运用纯无网格方法或粒子法对高维问题的计算模拟中,为得到较高精度数值结果常需要上百万离散节点或粒子数,这将会导致巨大存储或较长时间计算的问题^[15-20]。根据文献[15,17-18]知,对不含全局大型系数矩阵的纯无网格法或粒子法易于并行实施以提高计算效率。为此,已有许多基于 MPI 多 CPU 并行技术或算法被提出以加速纯无网格方法的计算^[17-18],为提高加速比就需要更多的 CPU,会使得计算硬件和服务器的管理成本增加。近些年来,随着计算机 GPU 的发展和 NVIDIA 公司不断更新的 CUDA 编程模型^[16,18-20],使得 GPU 能够进行大规模的并行计算,比如不涉及系数矩阵的纯无网格粒子法 (SPH 或耗散粒子动力学法 (DPD)) 的计算模拟已被成功地与 GPU 并行计算技术结合发展了高效的并行算法 (详见文献 [16,18-19,23]),且单个 GPU 的并行计算加速比可以达到 100 及以上 (与单个 CPU 串行计算相比),但一个 GPU 小工作站的管理成本远远低于有上百个 CPU 工作站的管理。

纯无网格 FPM 在高维多相分离过程的模拟还处在发展探索阶段,其并行算法实施的研究更是鲜有文献报道。本文首次考虑了基于 CUDA 的单个 GPU 来实现上述多元 C-H 方程 CFPM 离散格式的并行计算以提高效率,发展了 CFPM-GPU 并行算法。在 CFPM 离散格式的实施过程中涉及较大计算量或长时间计算的主要有两个方面:① 相邻节点的标定^[23],在时间层更新前每个节点 i (对应位置 \mathbf{x}_i , $i = 1, 2, \dots, N$, N 为离散节点数)都需要通过搜索所有离散节点确定其支持域内相邻节点 (每个点的相邻节点数不尽相同,其与离散点分布情况和位置有关系,本文数值模拟过程中离散点位置不变,故模拟实施中只需要循环一次相邻节点的标定),随着离散节点数的增加该标定的计算时间也会变大 (值得注意的是,本文数值模拟中节点或粒子位置在计算中固定);② 在每个时间层的物理量更新中, N 个离散点处空间导数的计算以及函数值的更新在串行编程实施过程中需要循环 N 次,包括局部矩阵的形成和求解,随着离散点数 N 的增大将会导致物理量每更新一次将需要较长时间的计算,但该循环可以同时并发进行 (完全不依赖网格方法的特点)。鉴于此,并行算法思想是重点对相邻节点编号标定和每次物理量更新中每个离散点处导数模型的循环实施并发性的计算。为实施 CFPM 离散格式的 GPU 并行计算 (并行算法较单个 CPU 的加速比可见 4.3 小节的讨论),本文采用基于 CPU 和 GPU 的异构计算框架,其中以 CPU 为主处理器 (程序中常用 host 表示) 和一个 GPU 为并行性计算实施的协处理器 (程序中常用 device 表示),具体实施以 CUDA-C++ 编程语言为主。另外需要注意的是,CFPM 格式中含 9×9 局部矩阵的形成和求解,其局部矩阵形成与相邻节点有关,在单个 CPU 上串行计算实施时可以采用向量 vector 的动态方式定义二维数组,准确地标记每个离散点处的相邻节点编号及相邻节点数,则在物理量更新计算的循环语句中形成局部矩阵时也需要用动态指针定义方式。但在 GPU 并行计算实施中发现,将上述串行实施采用的动态指针方式来定义计算循环里的局部矩阵会使得 GPU 内存不足,导致程序计算运行的阻塞。因此,在 GPU 并行算法编程中需要首先用常量的方式来给出最大相邻节点数 (估计值),然后动态定义一个一维或二维数组来标记相邻节点编号,同时在物理量更新计算的循环语句中可用静态方式来定义局部矩阵的形成。

一个离散模型需消耗 CPU 计算的实际时间,除了与编程语言和算法流程实施有一定关系外,受计算硬件性能指标的影响也很明显,本文数值模拟中采用的并行服务器工作站的主要硬件参数为: Intel Xeon Silver 4214 型号的 CPU 处理器两颗,每颗 CPU 处理器 12 核心 24 线程,主频为 2.2 GHz,动态加速频率为 3.2 GHz;三星 DDR4-2666 的内存共 128 GB;一个 NVIDIA Geforce Titan RTX 型号的 GPU,其显存容量为 24 GB,核心频率 1 770 MHz, CUDA 核心数 4 608 个。本文给出的 CFPM-GPU 并行算法流程可简化为如下几个步骤:

1) 选定光滑长度 h 和时间步长 dt ,并用常量定义方式给计算中需要的一些参数变量赋值。值得注意的是,考虑计算区域 Ω 为长方形或长方体 (相分离发生在一个复杂非规则区域 $\tilde{\Omega}$ 内且 $\tilde{\Omega} \subset \Omega$),对 Ω 进行离散得节点数为 N ,其中 $\Omega, \tilde{\Omega}$ 上离散点采用均匀分布情况, $\tilde{\Omega}$ 上离散节点可根据其形状特点进行非均匀布点方式,但相邻两个节点的距离近似等于均匀布点的情况 (比如圆形或球体上可以采用圆形或球形方式布点,也体现了本文采用的纯无网格法可以任意布点的优点)。

2) 动态方式定义所需的变量分配相关变量的 host 内存及数据初始化, 其中对函数 ϕ_3 赋初值 (计算中不变), 函数 ϕ_1 和 ϕ_2 赋初始时刻的值; 分配 device 内存, 并把 host 上数据拷贝到 device 上 (CPU 到 GPU); 调用 CUDA 核函数在 device 上并行方式运算标记出相邻节点的编号, 便于后面物理量随时间更新循环运算。

3) 当 $N_{\text{num}}=1$ 时 (设 N_{num} 为计算时间层数, 实际时间 $t=dt \times N_{\text{num}}$), 调用两次 CUDA 核函数在 device 上并行实现方程 (5) 右端项 CFPM 格式的运算, 并结合时间项格式的第一步, 得到每个离散点处函数 $\phi_{1,i}$ 的第一个中间值 $\phi_{1,i}^*$, 其中包括局部矩阵形成和调用在 device 与 host 上都编译的求解线性方程组的子函数。

4) 结合时间项格式的第二步重复步骤 3) 的运算, 将得到每个离散点处函数 $\phi_{1,i}$ 的第二个中间值 $\phi_{1,i}^{**}$ 。

5) 通过上述得到的两个中间值 $\phi_{1,i}^*$ 和 $\phi_{1,i}^{**}$, 结合时间项上采用的预估校正格式的第三步, 调用 CUDA 核函数在 device (GPU) 上并行得到 $N_{\text{num}}=1$ 时间层上每个离散点处第一个函数值 $\phi_{1,i}^1$; 然后把 device 上的计算结果传递到 host 上 (GPU 上信息传到 CPU 上)。

6) 运用式 (3) 在 host (CPU) 上计算得到 $N_{\text{num}}=1$ 时间层上每个离散点处第二个函数值 $\phi_{2,i}^1$, 然后再把 host 上结果传递到 device 上。

7) 重复步骤 3)~6) 可以得到下一个时间层上每个离散点上的函数值 $\phi_{1,i}$ 和 $\phi_{2,i}$ 。

8) 根据需要输出指定时间层上的计算结果。

3 CFPM-GPU 并行算法有效性验证

文献 [24] 主要对 FPM 离散格式求解二维单分量 C-H 方程的数值收敛性进行了分析, 但未对多分量 C-H 描述的明显相分离过程以及三维 C-H 方程进行数值研究. 本节采用文献 [8] 中具有二维圆环和三维球壳收缩相分离过程的两个基准算例, 对给出的 CFPM-GPU 算法求解带 Neumann 边界二维/三维 C-H 方程的准确性和高效性进行分析, 从而验证本文算法捕捉相分离过程的有效性。

文献 [8] 中两个相分离的基准算例没有理论解析解, 其通过坐标变换给出了高精度的数值解作为基准解, 并列出了某一时刻 64 个网格点上的基准解 (详见文献 [8] 中表 1 和表 3) 用来进行误差分析. 为了体现本文算法的准确性, 参照文献 [8] 中误差分析方式, 则有:

二范数误差

$$E_2 = \sqrt{2(d-1)\pi \sum_{i=1}^{64} (e_i)^2 (x_i)^{d-1} d_0} \quad (8)$$

和最大范数误差

$$E_\infty = \max_{1 \leq i \leq 64} |e_i|, \quad (9)$$

其中, d 为空间维数, $e_i = \phi_{64+i,65}^N - \phi_i^{\text{bs}}, i = 1, 2, \dots, 64$, $\phi_{64+i,65}^N$ 和 ϕ_i^{bs} 表示本文方法数值解和文献 [8] 中的基准解。

本节数值模拟二维圆环和三维球壳收缩过程对应的 C-H 方程可详见文献 [8], 其中势函数 $F(\phi) = 0.25(\phi^2 - 1)^2$, 梯度能量系数 $\varepsilon = 8d_0/[2\sqrt{2}\tanh^{-1}(0.9)]$, 值得注意的是, 为与文献 [8] 中的基准解做对比, 本节数值模拟中的离散点均采用的是均匀分布方式, 且离散点间距 (空间步长) $d_0 = 1/64$ 。

3.1 二维圆环收缩过程模拟

考虑二维空间区域 $\Omega = (-1, 1)^2$ 下圆环收缩现象, 并将区域均匀离散成 $N = 129 \times 129$ 个点, 对应节点间距 $d_0 = 1/64$, 时间步长取 $dt = 50d_0^4$, 运算结束时间取 $T = 8\,000\,000dt \approx 2.384$. 选取初始时刻函数:

$$\phi(x, y, 0) = \tanh \left(\frac{0.1 - \left| \sqrt{x^2 + (y - 0.5d_0)^2} - 0.75 \right|}{\sqrt{2}\varepsilon} \right). \quad (10)$$

经物理量更新循环 8 000 000 时间步运算, 本文 CFPM-GPU 法得到的数值解与基准解 (见文献 [8] 中表 1) 的二范数误差 $E_2 = 0.003\,39$, 最大范数误差 $E_\infty = 0.004\,96$. 上述表明, 本文方法求解二维圆环收缩过程对应的 C-H 方程是准确稳定的, 得到的圆环收缩现象也是可靠的. 图 1 给出了三个时刻二维圆环收缩现象的 CFPM-

GPU 数值结果, 其与文献 [8] 中图 6 数值结果吻合. 从图 1 中可以看出圆环外圆与内圆随时间延长逐渐缩小形成圆环的收缩过程, 结合上述算法误差精度可知, 圆环的内外圆半径随时间延长的变小过程与文献 [8] 的基准解一致, 从而体现本文给出的 CFPM-GPU 法模拟二维情况下相分离过程是准确有效的.

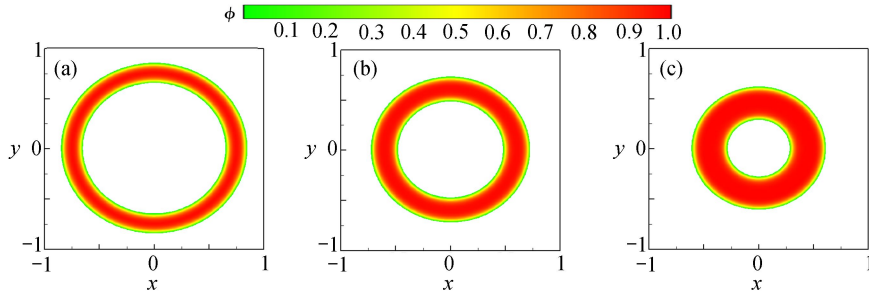


图 1 二维圆环收缩现象 CFPM-GPU 数值模拟结果: (a) $t=0$; (b) $t=1.31$; (c) $t=2.38$

Fig. 1 The 2D shrinking annuli obtained with the CFPM-GPU: (a) $t=0$; (b) $t=1.31$; (c) $t=2.38$

注 为了解释图中的颜色, 读者可以参考本文的电子网页版本, 后同.

3.2 三维球壳收缩过程模拟

三维球壳收缩过程对应的是球对称的 C-H 方程, 数值模拟中考虑三维空间域 $\Omega = (-1, 1)^3$, 并将区域均匀离散成 $N = 129 \times 129 \times 129$ 个点, 对应节点间距 $d_0 = 1/64$, 时间步长取 $dt = 50d_0^4$, 运算结束时间取 $T = 400\,000dt \approx 1.192$. 选取初始时刻函数:

$$\phi(x, y, z, 0) = \tanh \left(\frac{0.1 - \sqrt{x^2 + (y - 0.5h)^2 + (z - 0.5h)^2} - 0.75}{\sqrt{2}\varepsilon} \right). \tag{11}$$

经 400 000 时间步运算, CFPM-GPU 法得到 $t=1.192$ 时刻的三维球壳收缩现象, 并与初始时刻结果做对比 (见图 2), 图 2 给出了两个时刻三维球壳收缩形状和二维截面圆环, 其展现出明显的三维球壳收缩现象与文献 [8] 中图 10 结果吻合. 为体现本文方法数值求解的准确性, 将沿半圆环线上的数值结果与基准解 (见文献 [8] 中表 3) 进行对比, 得到二范数误差 $E_2 = 0.005\,4$ 和最大范数误差 $E_\infty = 0.007\,81$. 通过上述数值误差和三维球壳收缩现象表明本文方法模拟三维相分离过程准确.

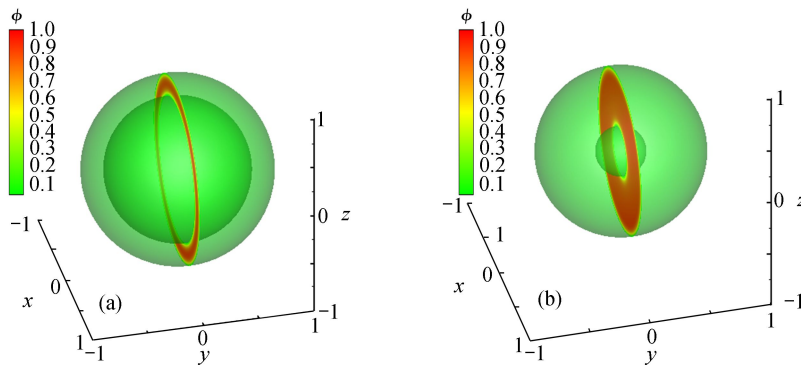


图 2 三维球壳收缩现象 CFPM-GPU 数值模拟结果: (a) $t=0$; (b) $t=1.192$

Fig. 2 The 3D spherical shells obtained with the CFPM-GPU: (a) $t=0$; (b) $t=1.192$

3.3 GPU 并行算法加速比分析

为体现本文基于 CUDA 的单个 GPU 并行算法的高效性, 将其与单个 CPU 的串行计算时间进行对比, 关于 GPU 并行服务器的性能参数见 2.2 小节. 为体现并行算法的效果, 采用 3.2 小节三维 C-H 方程的算例, 并定义加速比:

$$\delta_{\text{Sur}} = t_C / t_{C-G}, \tag{12}$$

其中, t_C 表示单个 CPU 计算消耗的时间, t_{C-G} 表示单个 CPU-GPU 异构并行计算消耗的时间, δ_{Sur} 为加速比.

根据 2.2 小节里并行算法思想的讨论, 需要并行加速提高计算效率的有相邻节点标定和物理量更新循环

两个部分, 表 1 和表 2 分别给出了相邻节点标记消耗计算时间和每个时间层里物理量更新循环所需平均计算时间及加速比. 由表 1 和表 2 知, 随节点数增加, 单个 CPU 计算时间和 GPU 并行计算时间也都随之非线性增加, 但 GPU 并行消耗计算时间远少于 CPU 计算时间, 表明基于 CPU-GPU 异构并行计算较单个 CPU 串行提高运算效率是显著的, 且 GPU 并行算法的加速比约为 160.

表 1 不同节点数下, 相邻节点标定消耗计算时间对比

Table 1 The consumed computing time for the calibration of neighbor nodes under different node numbers

node number	computing time		
	CPU T_{CPU}/s	GPU T_{GPU}/s	δ_{Sur}
$65 \times 65 \times 65$	833.55	4.73	176.2
$129 \times 129 \times 129$	51 513.65	287.69	179.1
$257 \times 257 \times 257$	3 226 951.32	17 907.61	180.2

表 2 不同节点数下, 每个时间层里物理量更新循环所需平均计算时间对比

Table 2 The average computing time costs at each time step under different node numbers

node number	computing time		
	CPU T_{CPU}/s	GPU T_{GPU}/s	δ_{Sur}
$65 \times 65 \times 65$	10.332	0.065	158.95
$129 \times 129 \times 129$	86.311	0.539	160.13
$257 \times 257 \times 257$	725.281	4.525	160.28

4 二维/三维多相分离过程数值预测

为进一步体现本文给出的 CFPM-GPU 并行算法数值模拟高维复杂区域上多相分离现象 (m 相分离情况对应 $m+1$ 元 C-H 方程), 本节选取 5 个代表性的复杂区域上的相分离算例^[10], 其中包含 3 个二维算例 (圆盘域内和脑切面域内二相分离, 星形域内三相分离) 和 2 个三维算例.

4.1 二维多相分离现象数值预测

算例 1 二维圆盘域上二相分离现象

考虑二维圆盘域 $\tilde{\Omega} = \{(x, y) | \sqrt{x^2 + y^2} < 0.45\}$ 上的二相分离现象, 计算区域取单位正方形 $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$, $\tilde{\Omega}$ 嵌入在 Ω 中且两个区域中心相同. 数值模拟中三元 C-H 方程中初始条件为

$$\phi_3(x, y) = 0.5 + 0.5 \tanh\left(\frac{\sqrt{x^2 + y^2} - 0.45}{2\sqrt{2}\varepsilon}\right), \quad \phi_1(x, y, 0) = [1 - \phi_3(x, y)][0.5 + 0.5\text{rand}(x, y)],$$

$$\phi_2(x, y, 0) = 1 - \phi_1(x, y, 0) - \phi_3(x, y),$$

其中 $\text{rand}(x, y)$ 表示介于 -1 和 1 之间的随机数. 选取节点间距为 $d_0 = 1/512$, 时间步长为 $dt = 10^{-6}$. 运算到 10^5 时间步 (对应时间 $t = 10^5 dt$) 结束, 图 3 给出了几个时刻下圆盘域上二相分离现象的 CFPM-GPU 模拟结果, 其中蓝色为 ϕ_1 , 红色为 ϕ_2 . 通过图 3 可观察到在较短时间里 (约 $t=0.05$) 就出现了明显的相分离现象, 且随时间延长相分离越来越明显. 本文结果与文献 [10] 中图 3 的多重网格法得到的二相分离过程趋势一致, 表明本文 CFPM-GPU 法模拟预测二相分离现象是可靠的 (注意, 两者间存在的差异主要由随机数 $\text{rand}(x, y)$ 生成的初始值不完全相同导致).

算例 2 二维脑剖面域上二相分离现象

为进一步体现本文方法模拟较复杂区域上二相分离现象的能力, 将文献 [10] 中二维脑剖面轮廓形成区域视为发生二相分离的复杂区域, 并嵌入到单位计算区域 $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$ 内. 图 4 给出了二维脑剖面原图及复杂脑剖面轮廓作为区域 $\tilde{\Omega}$, 并通过图像处理技术和迭代计算得到模拟的初始值 ($\tilde{\Omega} = \{(x, y) | \phi_3(x, y) < 0.5\}$). 具体步骤如下: 第一步, 将脑剖面轮廓图 4(b) 拷贝到画图工具里, 设置像素大小为 512×512 ; 第二步, 用 MATLAB 处理图像命令读取第一步里的 RGB 像素矩阵, 并输出到一个 txt 文件里; 第三步, 通过编程对第二步得到的数据进行 0-1 化处理, 得到另一个 txt 文件, 将其作为模拟初始数据的估计值; 第四步, 将第三步里得到的数据代入到 C-H 方程中, 经过 10 次迭代运算得到的值作为模拟初始的数据. 模拟

中初始时刻函数值除了 ϕ_3 ,其他两个与上述算例1的相同.对单位计算区域采用离散节点 513×513 ,时间步长 $dt = 10^{-6}$,经 10^5 时间步运算结束.图5给出了CFPM-GPU并行算法模拟的二维复杂脑剖面域上随时间演化的二相分离过程,其中蓝色为 ϕ_1 ,红色为 ϕ_2 .通过图5可知,本文方法能够模拟复杂区域上的二相分离现象,且与文献[10]中图5结果趋势一致(两者存在差异的主要原因在于随机数产生的初始值不完全相同),进一步表明CFPM-GPU模拟预测复杂区域上二相分离现象是有效的.

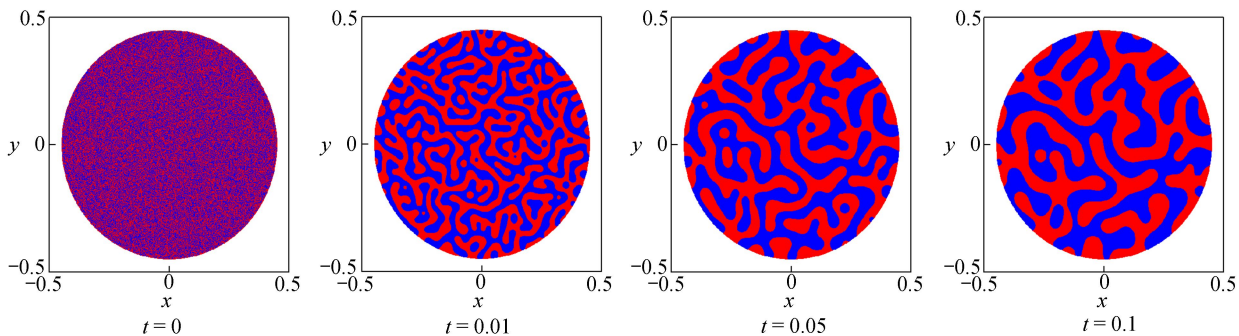


图3 二维圆盘域上二相分离现象CFPM-GPU模拟结果

Fig. 3 The 2-phase separation obtained with the CFPM-GPU in the 2D disk domain

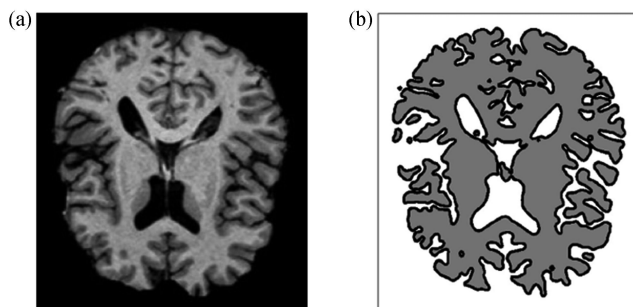


图4 二维脑剖面原图及其复杂脑剖面轮廓^[10]: (a)脑剖面原图; (b)脑剖面计算区域图

Fig. 4 The 2D brain section picture and the complex brain shape domain^[10]: (a) the brain section picture; (b) the brain section calculation domain

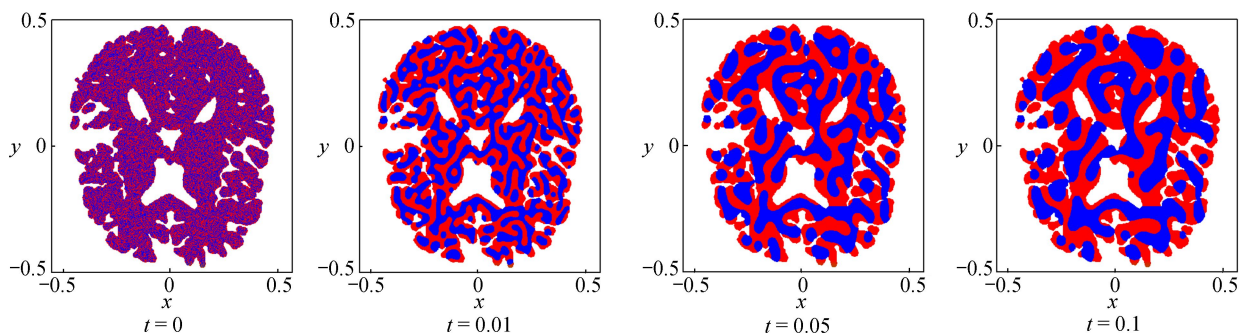


图5 二维脑剖面域上二相分离现象CFPM-GPU模拟结果

Fig. 5 The 2-phase separation obtained with the CFPM-GPU in the 2D brain shape domain

算例3 二维星形域上三相分离现象

许多物理问题系统中存在多组分的三相分离现象,比如多相合金和三元纳米线制备中^[28].因此,本节考虑了在单位计算区域 $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$ 内星形域上三相分离过程的CFPM-GPU法模拟,对应四元C-H系统的求解.四元混合物中各个分量 $\phi_i(x, y, t)$ ($i = 1, 2, 3, 4$)同样满足质量守恒律 $\phi_1 + \phi_2 + \phi_3 + \phi_4 = 1$,且星形域 $\tilde{\Omega} = \{(x, y) | \phi_4(x, y) < 0.5\}$,各个分量初始条件为

$$\phi_4(x, y) = 0.5 + 0.5 \tanh\left(\frac{\sqrt{x^2 + y^2} - 0.35 - 0.1 \cos(7\theta)}{2\sqrt{2}\varepsilon}\right),$$

$$\phi_1(x, y, 0) = [1 - \phi_4(x, y)][0.35 + 0.05\text{rand}(x, y)],$$

$$\begin{aligned} \phi_2(x, y, 0) &= [1 - \phi_4(x, y)][0.35 + 0.05\text{rand}(x, y)], \\ \phi_3(x, y, 0) &= 1 - \phi_1(x, y, 0) - \phi_2(x, y, 0) - \phi_4(x, y), \end{aligned}$$

其中 $\theta = \tan^{-1}(y/x)$ 且 $x \neq 0$, 随机数 $\text{rand}(x, y)$ 同上。值得注意的是, ϕ_1 和 ϕ_2 的初始值表达式相同, 但编程计算时使其取的随机数值不完全相同, 且计算中需要求解关于 ϕ_1 和 ϕ_2 的 C-H 方程组。采取的计算节点数与时间步长同上, 经 10^6 时间步运算得到几个不同时刻的三相分离现象 (见图 6)。观察图 6 可知, 在 $t=0.1$ 时刻出现了明显的三相分离现象, 随时间演化三相分离现象越来越清晰 (红色区域为 ϕ_1 , 蓝色区域为 ϕ_2 , 绿色区域为 ϕ_3), 表明本文方法模拟预测非规则区域上三相分离过程是有效的。

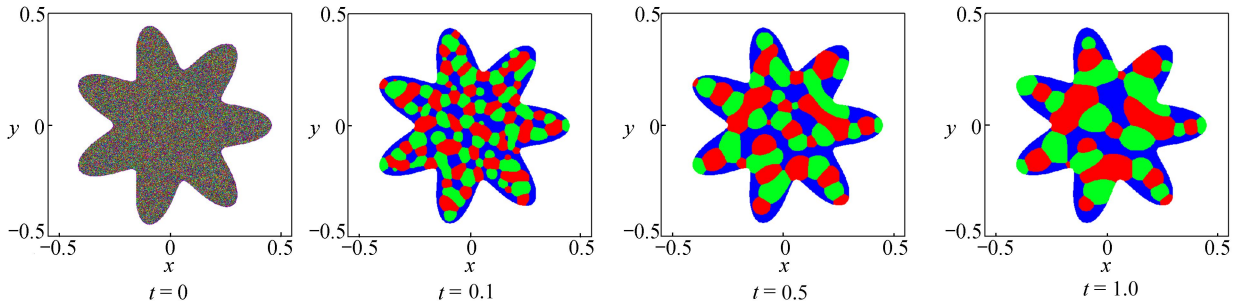


图 6 二维星形域上三相分离现象 CFPM-GPU 模拟结果

Fig. 6 The 3-phase separation obtained with the CFPM-GPU in the 2D star domain

4.2 三维多相分离现象数值预测

算例 4 三维环形体域上相分离现象

为体现 CFPM-GPU 法模拟三维复杂域上相分离现象的有效性, 考虑三维环形体域上二相分离, 取环体区域 $\tilde{\Omega} = \left\{ (x, y, z) \mid \sqrt{(\sqrt{x^2 + y^2} - 0.3)^2 + z^2} < 0.15 \right\}$ 嵌入在计算区域 $\Omega = (-0.5, 0.5) \times (-0.5, 0.5) \times (-0.25, 0.25)$ 内。三元 C-H 方程中每个分量的初始条件为

$$\phi_3(x, y, z) = 0.5 + 0.5 \tanh \left(\frac{\sqrt{(\sqrt{x^2 + y^2} - 0.3)^2 + z^2} - 0.15}{2\sqrt{2}\epsilon} \right),$$

$$\phi_1(x, y, z, 0) = [1 - \phi_3(x, y, z)][0.5 + 0.5\text{rand}(x, y, z)],$$

$$\phi_2(t=0) = 1 - \phi_1(t=0) - \phi_3,$$

其中随机数 $\text{rand}(x, y, z)$ 同上。

模拟中采取离散节点 $129 \times 129 \times 65$, 初始间距 $d_0 = 1/128$ 和时间步长为 $dt = 10^{-6}$ 。图 7 展示了本文数值方法得到的关于 ϕ_1 三个时刻的相分离现象, 在较短时间 $t=0.01$ 时就出现相分离, 随时间演化相分离现象越发明显, 且与文献 [10] 中图 7 结果趋势一致。

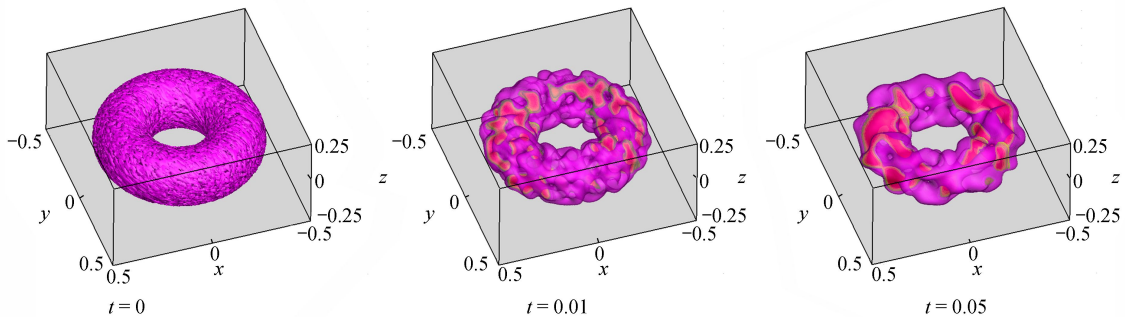


图 7 三维环形体域上相分离现象 CFPM-GPU 模拟结果

Fig. 7 The phase separation obtained with the CFPM-GPU in the 3D torus domain

算例 5 三维 Schwarz-D 域上相分离现象

为进一步体现 CFPM-GPU 法模拟三维复杂域上相分离现象的可靠性, 考虑 Schwarz-D 形域^[10] 嵌入在计

算域 $\Omega = (0, 1) \times (0, 1) \times (0, 1)$ 的二相分离, 其中发生相分离的复杂 Schwarz-D 形域 $\tilde{\Omega} = \{(x, y, z) | C(x, y, z) < 0\}$. 三元 C-H 方程中每个分量的初始条件为

$$\begin{aligned}\phi_3(x, y, z) &= 0.5 + 0.5 \tanh\left(\frac{C(x, y, z)}{20\sqrt{2}\varepsilon}\right), \\ \phi_1(x, y, z, 0) &= [1 - \phi_3(x, y, z)][0.5 + 0.5\text{rand}(x, y, z)], \\ \phi_2(x, y, z, 0) &= 1 - \phi_1(x, y, z, 0) - \phi_3(x, y, z),\end{aligned}$$

其中

$$C(x, y, z) = \cos(2\pi x)\cos(2\pi y)\cos(2\pi z) - \sin(2\pi x)\sin(2\pi y)\sin(2\pi z).$$

模拟中采取离散节点 $129 \times 129 \times 129$, 初始间距和时间步长与算例 4 相同. 图 8 展示了三维复杂 Schwarz-D 形域上 CFPM-GPU 法模拟的相分离现象, 可观察到在较短时间 $t=0.01$ 时就出现明显的相分离现象, 本文数值结果与文献 [10] 中的相分离趋势一致.

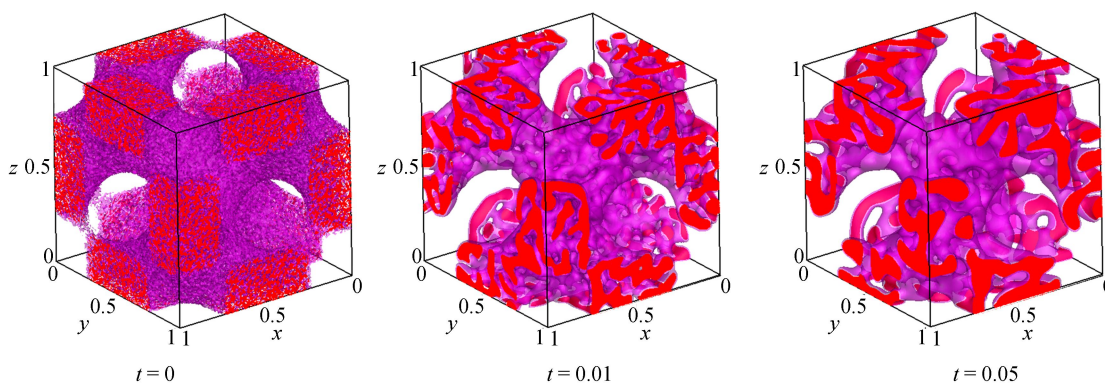


图 8 三维 Schwarz-D 区域上相分离现象 CFPM-GPU 模拟结果

Fig. 8 The phase separation obtained with the CFPM-GPU in the 3D Schwarz-D domain

通过算例 1~算例 5 二维/三维复杂域上相分离现象的 CFPM-GPU 模拟, 并与文献 [10] 中图 8(b) 结果比较可知: 本文提出的 CFPM-GPU 求解三元或四元 C-H 方程组是有效的; CFPM-GPU 数值预测二维/三维复杂域上的二相或三相分离现象是高效可靠的.

5 结 论

本文针对二维/三维复杂域上多相分离现象的高效性与准确性模拟, 首次结合 CPU-GPU 异构并行计算、拓展求解含四阶导数二维 C-H 方程的 FPM 和采用具有较好光滑性的 Wendland 权函数, 给出了一种基于 CUDA 语言单个 GPU 的 CFPM-GPU 并行算法. 为体现 CFPM-GPU 模拟预测复杂域上多相分离的有效性, 首先运用 CFPM-GPU 分别对两个相分离的基准算例 (二维/三维) 进行了模拟, 并对误差和并行加速效率进行了分析; 其次, 用 CFPM-GPU 对无解析解的多个二维/三维复杂域上的二相或三相分离现象进行了数值预测, 并与多重网格法结果比较. 所有数值结果表明:

- 1) 本文的 CFPM-GPU 并行算法求解二维/三维 C-H 方程描述的相分离过程是准确、高效的;
- 2) 基于 CUDA 的单个 GPU 并行较 CPU 串行运算的加速比约为 160 倍;
- 3) 给出的 CFPM-GPU 能准确预测二维/三维复杂区域上二相或三相分离现象随时间的演化过程.

致谢 本文作者衷心感谢扬州大学本科专业品牌建设与提升工程项目 (ZYPP2018B007) 对本文的资助.

参考文献 (References):

- [1] CHOKSI R, PELETIER M, WILLIAMS J F. On the phase diagram for microphase separation of diblock copolymers: an approach via a nonlocal Cahn-Hilliard functional[J]. *SIAM Journal on Applied Mathematics*, 2009, **69**(6): 1712-1738.
- [2] LI Y B, KIM J. Multiphase image segmentation using a phase-field model[J]. *Computers and Mathematics With*

- Applications*, 2011, **62**(2): 737-745.
- [3] 汪精英, 翟术英. 分数阶 Cahn-Hilliard 方程的高效数值算法[J]. 应用数学和力学, 2021, **42**(8): 832-840. (WANG Jingying, ZHAI Shuying. An efficient numerical algorithm for fractional Cahn-Hilliard equations[J]. *Applied Mathematics and Mechanics*, 2021, **42**(8): 832-840.(in Chinese))
- [4] CHARLES M E, HARALD G. Diffusional phase transitions in multicomponent systems with a concentration dependent mobility matrix[J]. *Physica D: Nonlinear Phenomena*, 1997, **109**(3/4): 242-256.
- [5] DIEGEL A, FENG X B, WISE S M. Analysis of a mixed finite element method for a Cahn-Hilliard-Darcy-Stokes system[J]. *SIAM Journal on Numerical Analysis*, 2015, **53**: 127-152.
- [6] GUO Z, LIN P, LOWENGRUB J, et al. Mass conservative and energy stable finite difference methods for the quasi-incompressible Navier-Stokes-Cahn-Hilliard system: primitive variable and projection type schemes[J]. *Computer Methods in Applied Mechanics and Engineering*, 2017, **326**: 144-174.
- [7] ZHANG Z G, QIAO Z H. An adaptive time-stepping strategy for the Cahn-Hilliard equation[J]. *Communications in Computational Physics*, 2012, **11**(4): 1261-1278.
- [8] JEONG D, CHOI Y, KIM J. A benchmark problem for the two- and three-dimensional Cahn-Hilliard equation[J]. *Communications in Nonlinear Science and Numerical Simulation*, 2018, **61**: 149-159.
- [9] KIM J. A numerical method for the Cahn-Hilliard equation with a variable mobility[J]. *Communications in Nonlinear Science and Numerical Simulation*, 2007, **12**: 1560-1571.
- [10] JEONG D, YANG J X, KIM J. A practical and efficient numerical method for the Cahn-Hilliard equation in complex domains[J]. *Communications in Nonlinear Science and Numerical Simulation*, 2019, **73**: 217-228.
- [11] CHENG R J, CHENG Y M. Solving unsteady Schrödinger equation using the improved element-free Galerkin method[J]. *Chinese Physics B*, 2016, **25**: 020203.
- [12] DEHGHAN M, ABBASZADEH M. The meshless local collocation method for solving multi-dimensional Cahn-Hilliard, Swift-Hohenberg and phase field crystal equations[J]. *Engineering Analysis With Boundary Elements*, 2017, **78**: 49-64.
- [13] 王红, 李小林. 二维瞬态热传导问题的无单元 Galerkin 法分析[J]. 应用数学和力学, 2021, **42**(5): 460-469. (WANG Hong, LI Xiaolin. Analysis of 2D transient heat conduction problems with the element-free Galerkin method[J]. *Applied Mathematics and Mechanics*, 2021, **42**(5): 460-469.(in Chinese))
- [14] 曹维鸿, 傅卓佳, 汤卓超. 水槽动力特性数值模拟的新型局部无网格配点法[J]. 应用数学和力学, 2022, **43**(4): 392-400. (CAO Weihong, FU Zhuojia, TANG Zhuochao. A novel localized meshless collocation method for numerical simulation of flume dynamic characteristics[J]. *Applied Mathematics and Mechanics*, 2022, **43**(4): 392-400.(in Chinese))
- [15] LIU G R, LIU M B, LI S. Smoothed particle hydrodynamics: a mesh-free particle method[J]. *Computational Mechanics*, 2003, **33**: 491.
- [16] 林晨森, 陈硕, 李启良, 等. 耗散粒子动力学 GPU 并行计算研究[J]. 物理学报, 2014, **63**(10): 104702. (LIN Chensen, CHEN Shuo, LI Qiliang, et al. Accelerating dissipative particle dynamics with graphic processing unit[J]. *Acta Physica Sinica*, 2014, **63**(10): 104702.(in Chinese))
- [17] JIANG T, CHEN Z C, LU W G, et al. An efficient split-step and implicit pure mesh-free method for the 2D/3D nonlinear Gross-Pitaevskii equations[J]. *Computer Physics Communications*, 2018, **231**: 19-30.
- [18] CRESPO A J C, DOMÍNGUEZ J M, ROGERS B D, et al. DualSPHysics: open-source parallel CFD solver based on smoothed particle hydrodynamics (SPH)[J]. *Computer Physics Communications*, 2015, **187**: 204-216.
- [19] KING J R, POGORELOV I V, AMYX K M, et al. GPU acceleration and performance of the particle-beam-dynamics code Elegant[J]. *Computer Physics Communications*, 2019, **235**: 346-355.
- [20] HE F, ZHANG H S, HUANG C, et al. A stable SPH model with large CFL numbers for multi-phase flows with large density ratios[J]. *Journal of Computational Physics*, 2022, **453**: 110944.
- [21] 杨秀峰, 刘谋斌. 瑞利-泰勒不稳定问题的光滑粒子法模拟研究[J]. 物理学报, 2017, **66**(16): 164071. (YANG Xiufeng, LIU Moubin. Numerical study of Rayleigh-Taylor instability by using smoothed particle hydrodynamics [J]. *Acta Physica Sinica*, 2017, **66**(16): 164071.(in Chinese))

- [22] HE F, ZHANG H S, HUANG C, et al. Numerical investigation of the solitary wave breaking over a slope by using the finite particle method[J]. *Coastal Engineering*, 2020, **156**: 103617.
- [23] DANIEL W, MASSOUD R, WOLFGANG R. Neighbour lists for smoothed particle hydrodynamics on GPUs[J]. *Computer Physics Communications*, 2018, **225**: 140-148.
- [24] 任金莲, 蒋戎戎, 陆伟刚, 等. 基于局部加密纯无网格法非线性Cahn-Hilliard方程的模拟[J]. *物理学报*, 2020, **69**(8): 080202. (REN Jinlian, JIANG Rongrong, LU Weigang, et al. Simulation of nonlinear Cahn-Hilliard equation based on local refinement pure meshless method[J]. *Acta Physics Sinica*, 2020, **69**(8): 080202.(in Chinese))
- [25] SUCHDE P, KUHNERT J, TIWARI S. On meshfree GFDM solvers for the incompressible Navier-Stokes equations[J]. *Computers & Fluids*, 2018, **165**: 1-12.
- [26] SUDARSHAN T, JÖRG K. Modeling of two-phase flows with surface tension by finite pointset method (FPM)[J]. *Journal of Computational and Applied Mathematics*, 2007, **203**: 376-386.
- [27] EDGAR O R F, IRMA D G C. Application of the finite pointset method to non-stationary heat conduction problems[J]. *International Journal of Heat and Mass Transfer*, 2014, **71**: 720-723.
- [28] LI Y B, CHOI J I, KIM J. Multi-component Cahn-Hilliard system with different boundary conditions in complex domains[J]. *Journal of Computational Physics*, 2016, **323**: 1-16.