2021 年 6 月 数值计算与计算机应用 第 42 卷第 2 期
Jun., 2021 Journal on Numerical Methods and Computer Applications Vol.42, No.2

doi: 10.12286/szjs.s2020-0714

基于 PHG 平台的非结构四面体网格欧拉方程间断有限元并行求解器*1)

杜玉龙
(北京航空航天大学数学科学学院, 北京 100091)

徐凯文 赵昆磊 袁 礼
(中国科学院数学与系统科学研究院计算数学所, 科学与工程计算国家重点实验室, 北京 100190;
中国科学院大学数学科学学院, 北京 100090)

摘 要

针对计算流体力学对高性能计算的需求, 基于三维并行自适应有限元程序开发平台 PHG (Parallel Hierarchical Grid) 开发了在非结构四面体网格上求解可压缩流欧拉方程的间断有限元法并行求解器 (Libdgphg 库). 该求解器以 C++ 函数库的形式实现数值方法中各项功能. 实施了模态基一次间断有限元, 采用低耗散的 MLP (Multi-dimensional Limiting Process) 限制器来抑制间断附近的数值振荡. 由于 MLP 限制器需要所有与当前单元共享顶点的邻近单元的信息, 模板较宽, 这给程序设计带来一定的困难. 我们通过引入辅助向量收集共享顶点的所有单元中的最大、最小单元积分平均值, 并归属到单元数据结构上, 从而利用 PHG 内在的通信机制实现 MPI 分区间的信息交换. 通过几个数值算例测试了 Libdgphg 库的数值结果以及并行性能. 算例表明: 该求解器能得到理论精度阶和较高分辨率, 同时有良好的并行性能, 在千核测试中可达到 60% 以上的并行效率, 可用于流体问题的大规模计算.

关键词: 双曲守恒律; 并行计算; PHG 平台; MLP 限制器; 间断有限元法.

MR (2010) 主题分类: 65D17.

A PARALLEL DGM SOLVER FOR THE EULER EQUATIONS ON UNSTRUCTURED TETRAHEDRAL GRIDS BASED ON THE TOOLBOX PHG

Du Yulong
(*School of Mathematical Sciences, Beihang University, Beijing 10091, China*)

Xu Kaiwen Zhao Kunlei Yuan Li
(*LSEC and Institute of Computational Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100090, China;*
School of Mathematics Sciences, University of Chinese Academy of Sciences, Beijing 100090, China)

Abstract

We develop a C++ library *libdgphg* for solving the Euler equations on unstructured tetrahedral grids with a discontinuous Galerkin method (DGM), based on the 3D finite

* 2020 年 11 月 8 日收到.

1) 基金项目: 国家自然科学基金 (91641107, 91852116, 12071470) 和工信部专项 (MJ-F-2012-04) 资助.

element code development toolbox PHG (Parallel Hierarchical Grid). The solver uses C++ classes for different modules of the DGM. We have implemented the modal base P^1 DG method. The multi-dimensional limiting process (MLP) technique is applied to suppress numerical oscillations near discontinuities. To overcome the drawback of the non-compact stencil of the MLP limiter, we develop a strategy in which an auxiliary vector is introduced to collect the max-min values from cell averages of all vertex-sharing neighbour cells. Then, we distribute the max-min values in the auxiliary vector to the element object of PHG. In this way, we can realize the MPI communication automatically through the neighbour interface of PHG. Finally, we show that the MLP limiter can obtain expected accuracy and superior resolution, and the solver can attain 60 % parallel efficiency on 1024 cores.

Keywords: Hyperbolic conservation law; Parallel computing; Multi-dimensional limiting process; Parallel hierarchial grid; Discontinuous Galerkin method.

2010 Mathematics Subject Classification: 65D17.

1. 引　　言

可压缩流欧拉方程的数值求解是计算流体力学 (CFD, Computational Fluid Dynamics) 的核心研究内容之一 [1–3]. 最近二十多年来的研究表明, 模拟复杂流动问题时高阶方法比低阶方法的计算效率和效果要优越得多. 发展高精度数值方法成为了目前CFD 的研究焦点. 其中间断有限元方法 (Discontinuous Galerkin Method, DGM) 尤其得到了CFD 研究者的广泛重视. DGM 保持了传统有限元方法 (FEM) 的优点, 融入了有限体积法 (FVM) 的数值通量和限制器等思想, 具有局部守恒、能处理间断、适合复杂几何区域、适合并行计算和h-p 自适应等优点 [4–8]. 与此同时很多工程应用的几何外形是很复杂的, 非结构网格的几何适应性远远优于结构网格, 且生成快速, 自动化程度高, 容易进行自适应加密 [9–15]. 因此基于非结构网格的DGM 具有重要的应用前景, 已经成为CFD 应用领域的热门课题 [16–18].

在有限体积法中发展起来的MLP (Multi-dimensional Limiting Process) [19–21] 限制器是对传统MUSCL 型限制器 [22–26] 的一个重要改进. MLP 限制器使数值解满足真正多维极大值原理 (MUSCL 型限制器只满足面邻居中的极大值原理, 这可能丢失多维问题的极值), 从而满足真正多维的局部极值递减原则 (Local Extremum Diminishing, LED) [19]. 因此, MLP 限制器可以更有效的抑制多维振荡, 捕捉更高分辨的多维波结构, 对于多维流动有更高的计算分辨率和准确度. 但是 MLP 限制器模板涉及所有与本单元共享顶点的邻居单元, 相比只用面邻居单元 (von Neumann 邻居单元) 的DGM 的模板要宽很多, 并行计算中增加通信复杂性, 在三维情形下该问题更为突出. 为避免MLP 对一些单元的重复访问, 本文设计了一种新的实现策略, 通过引入总顶点数大小的辅助向量, 一次遍历单元收集共顶点邻居单元均值中的最大、最小值, 并归属到单元的数据结构上. 这样在整个算法中只涉及面邻居单元信息, 从而保持了DGM 的模板紧凑性, 且能利用PHG 的通信接口, 实现PHG 的子网格 (MPI 分区) 之间的信息交换.

为了实现三维复杂区域上流动的大规模计算, 我们基于中科院计算数学所研制的三维并行自适应有限元软件开发平台PHG [27–30], 开发了间断有限元方法求解守恒律 (含欧拉方程) 的C++ 求解器 (Libdgphg). 该求解器利用PHG 提供的高效的并行自适应有限元加密功能, 尤其是隐藏并行实现细节的接口, 用C++ 面向对象编程实现间断有限元方法中常用的场变量、数值通量、限制器、时间推进法等主要库. Libdgphg 的设计目标是以函数库的形式提供一个功能可扩展的间断有限元流体解法器包, 使用户可以方便的添加新的子模块以实现新的功能

或策略. 本文介绍了Libdgphg 求解器, 并给出测试算例.

2. 守恒律间断有限元方法简介

我们考虑三维双曲守恒律方程

$$\frac{\partial U}{\partial t}+\nabla\cdot\mathbf{F}(U)=0, \tag{2.1}$$

其中$U(\mathbf{x},t)=(u_1,u_2,\cdots,u_m)^T\in\mathbb{R}^m$是定义在$\Omega\times[0,\infty]$上的解, $\mathbf{F}=(F_1(U),F_2(U),F_3(U))$是通量函数. Ω是求解区域, $\mathbf{x}=(x,y,z)\in\Omega$. 给定初始条件

$$U(\mathbf{x},0)=U_0(\mathbf{x}), \tag{2.2}$$

和适定的边界条件.

首先, 将整个求解区域Ω划分成非结构四面体单元 Ω_j 的集合$\mathcal{T}_h$, 即$\Omega=\bigcup\limits_{j=1}^{N_e}\Omega_j,\Omega_j\in\mathcal{T}_h$, 其中$N_e$是网格单元个数. 设$P^k(\Omega_j)$是单元$\Omega_j$上的局部有限元空间, 取作$k\geq 0$次多项式集合(本文只做到$P^1$ 次多项式). 则在间断有限元空间

$$V_h=\{v_h\in L^\infty: v_h|_{\Omega_k}\in P^k(\Omega_j),\forall\Omega_j\in\mathcal{T}_h\} \tag{2.3}$$

中有限元近似解为

$$U_h(\mathbf{x},t)=\sum_{l=1}^{N_b}U_l^j(t)v_l^j(\mathbf{x}),\quad \mathbf{x}\in\Omega_j, \tag{2.4}$$

其中$v_l^j(\mathbf{x})\in V_h$是基函数, N_b表示基函数的个数 (本文中$N_b=4$).

其次, 令测试函数空间等于解函数空间, 将方程(2.1)乘以所有测试函数$v_l^j(\mathbf{x})\in V_h$并用要寻找的近似解$U_h(\mathbf{x},t)$替代精确解, 在单元Ω_j上积分, 由高斯积分公式得到弱形式

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_j}U_hv_l^j\mathrm{d}\Omega_j-\int_{\Omega_j}\nabla v_l^j\cdot\mathbf{F}(U_h)\mathrm{d}\Omega_j+\oint_{\partial\Omega_j}v_l^j\widehat{\mathbf{F}(U_h)\cdot\mathbf{n}}\mathrm{d}s=0,\quad j=1,2,\cdots,N_b, \tag{2.5}$$

其中$\mathbf{n}=(n_x,n_y,n_z)$和$\widehat{\mathbf{F}\cdot\mathbf{n}}=\hat{F}(U_h^-,U_h^+,\mathbf{n})$分别是单元$\Omega_j$界面单位外法向和数值通量. 本文Libdgphg库能够实现LF、HLLC、AUSM、Roe等常用数值通量格式.

最后, 所有单元上的半离散系统(2.5)可以写成半离散常微分方程组

$$\mathcal{M}_j\frac{\mathrm{d}\mathcal{U}_j(t)}{\mathrm{d}t}=\mathcal{R}_j\left(U_h(t)\right),\quad j=1,\cdots,N_e, \tag{2.6}$$

其中$\mathcal{U}_j(t)=(U_1^j(t),\cdots,U_{N_b}^j(t))^T$, $\mathcal{M}_j$为单元Ω_j上的质量矩阵:

$$\mathcal{M}_j=\int_{\Omega_j}V^T(\mathbf{x})V(\mathbf{x})\mathrm{d}\Omega_j,\quad V(\mathbf{x})=\left(v_1^j(\mathbf{x}),\cdots,v_{N_b}^k(\mathbf{x})\right). \tag{2.7}$$

将常微分方程组(2.6)式用时间格式离散求解, 就得到式(2.1)的数值解. 本文在Libdgphg库中我们采用具有 SSP (strong-stability-preserving)性质的1-3阶Runge-Kutta 方法 [31,32].

3. 基于并行层次网格 (PHG) 平台的 Libdgphg 简介

PHG 的主要功能是提供了许多方便用户编写并行自适应有限元程序的 C 语言数据结构和函数接口 [27,28]. 用户通过这些数据结构和函数接口, 可自动对网格进行分区、动态负载平衡和局部自适应加密, 同时方便的管理自由度数据、基函数、数值积分、求解线性方程组和特征值、计算后验误差估计子、计算结果可视化输出等. PHG 隐藏网格及并行细节 [29,30], 但提供透明的动态负载平衡功能, 用户只需调用平台提供的动态负载平衡函数, 便可自动完成网格重新划分、单元和用户数据的迁移以及新子网格重构.

我们基于 PHG 程序平台开发了用间断有限元法求解守恒律 (组) 的 Libdgphg 库, 其目标是在大规模四面体网格上计算双曲守恒律方程. Libdgphg 库采用 C++ 语言编写, 不同守恒律方程、数值通量、时间离散方法等策略均由不同的类表达. 按照 C++ 的常用设计原则, 类可以分为 "接口" 与 "实现". 对于只关心使用, 而不想实现新策略的使用者, 可以简单通过不同的类名称选择不同的策略, 即只需关心 "实现" 类. 而对于想自己定义新策略的用户, 则还需了解 "接口" 类 (虚类), 使新定义的策略符合 Libdgphg 的接口, 从而将新策略整合入 Libdgphg 库.

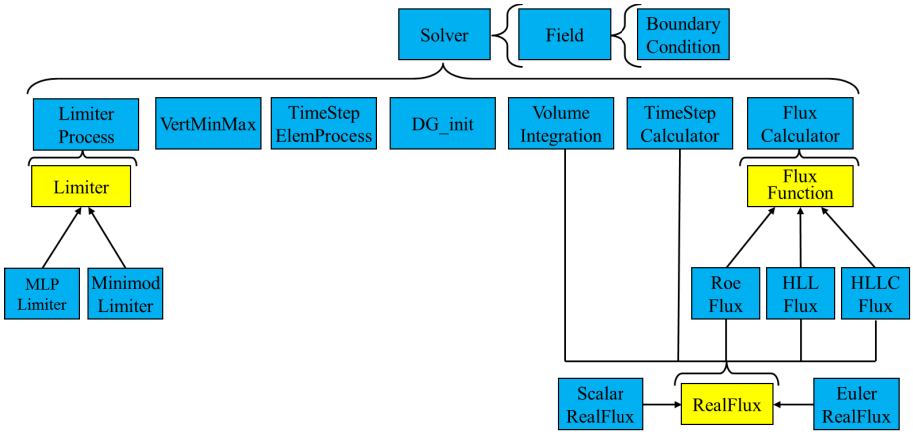

图 1 Libdgphg 库主要类关系图. 黄色框表示虚类, 蓝色框代表实现类. A→B 表示 A 继承 B, "B}A" 表示 A 包含 B

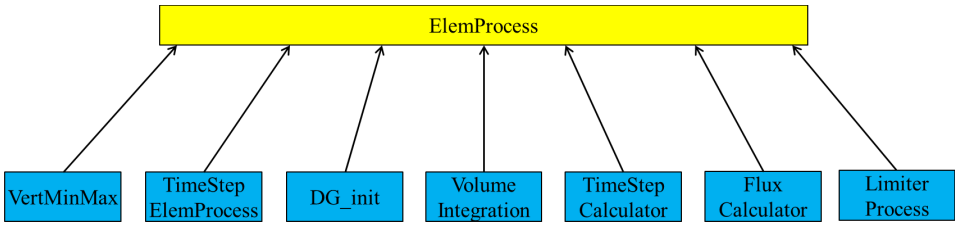

图 2 Libdgphg 库的单元遍历接口类和各个实现类关系图

Libdgphg 库的主要类关系如图 1 所示. 其中, 黄色矩形表示虚类, 即类定义中含有纯虚拟函数, 蓝色矩形表示实现类. 箭头线表示继承关系, 大括号表示包含关系. 图 2 展示了 Libdgphg 库用于单元遍历的接口类和各个实现类. 下面给出各个类主要功能的介绍.

Field 定义了求解区域上的变量场，是对 PHG 的网格 (grid) 相关的部分功能的封装. 该类提供单元上自由度、单元边界的数值通量、相邻单元的自由度等数据的访问接口, 同时包含边界条件类 BoundaryCondition, 处理给定边界条件.

Solver 是 libdgphg 库的核心, 实现一个时间步长的求解过程, 其核心是 Solver::TimeStep 函数. 类中成员包含: 变量场 Field 类, DG 自由度初始化过程 (DG_init 实现类), 限制过程 (LimiterProcess 实现类), 数值通量计算器 (FluxCalculator 实现类), 时间步长计算器 (TimeStepCalculator 实现类), 单元体积分计算器 (VolumeIntegration), Runge-Kutta 时间推进格式 (TimeStepElemProcess 实现类) 等 (见图 1).

ElemProcess 虚类, 为遍历所有单元的接口 (见图 2). 继承它的子类有：DG 自由度初始化过程 (DG_init 实现类), 限制过程 (LimiterProcess 实现类), 数值通量计算器 (FluxCalculator 实现类), 时间步长计算器 (TimeStepCalculator 实现类), 单元体积分计算器 (VolumeIntegration), Runge-Kutta 时间推进格式 (TimeStepElemProcess 实现类) 等. 该类提供虚拟函数 Do, 其子类在 Do 函数中实现在每个单元上执行的操作.

BoundaryCondition 为边界条件实现类, 可获取边界通量值. 按不同类型边界条件赋予单元边界面不同的通量值.

Limiter 虚类, 为不同的限制函数策略提供接口. 子类包括实现类 MLPLimiter 等.

FluxFunction 虚类, 为不同的数值通量函数提供接口. 其子类包括实现类 HLLCFlux, HLLFlux, RoeFlux 等, 可以计算不同的数值通量.

RealFlux 虚类, 为不同方程的通量提供接口, 子类包含 Euler RealFlux 和 Scalar RealFlux 分别实现 Euler 方程组和标量守恒律方程的理论通量的计算.

VertMinMax 实现类, 是虚类 ElemProcess 的子类, 其 Do 函数实现 MLP 限制器中辅助数组 $V_{\max}^{\min}$ 的遍历更新.

4. MLP 斜率限制器及实现的改进策略

多维情形下流场的信息可能按任意方向传播, 为更好的刻画多维流场性质, 本文的模态基函数线性元 DG 解的斜率采用文献 [20] 中的 MLP 斜率限制器. MLP 模板为包含所有与目标单元共享顶点的单元，见图 3 (为清楚起见，图示以二维三角网格为例).

考虑到线性元解的最大、最小值只能出现在单元顶点上，因此只限制单元顶点处的解值即可. 记 $\mathcal{V}(\Omega_j)$ 为单元 Ω_j 上所有顶点的集合, $\tilde{u}_j(\mathbf{x})$ 是未限制守恒变量分量的数值解, $\overline{\tilde{u}}_j(x) = \bar{u}_j^n + \alpha\nabla u_h^n \cdot (\mathbf{x} - \mathbf{x}_j)$ 是限制后的解. 则 MLP 限制因子 α 计算如下

$$\alpha_j = \min_{\forall v_i \in \mathcal{V}(\Omega_j)} \begin{cases} \min\left(1, \dfrac{\bar{u}_{v_i}^{\max} - \bar{u}_j^n}{\tilde{u}_{j,v_i} - \bar{u}_j^n}\right), & \text{if } \tilde{u}_{j,v_i} > \bar{u}_j^n \\ \min\left(1, \dfrac{\bar{u}_{v_i}^{\min} - \bar{u}_j^n}{\tilde{u}_{j,v_i} - \bar{u}_j^n}\right), & \text{if } \tilde{u}_{j,v_i} < \bar{u}_j^n \\ 1 & \text{otherwise} \end{cases}, \tag{4.1}$$

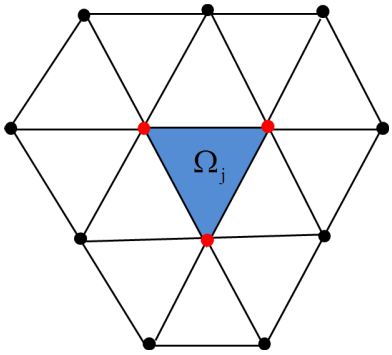

图 3 三角形网格 MLP 模板

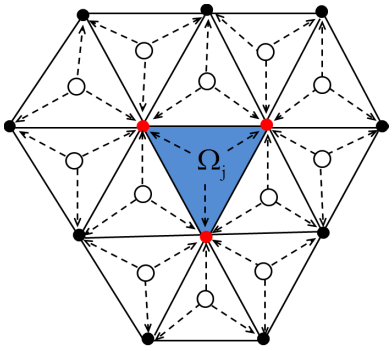

图 4 MLP 顶点信息归集

其中 $\tilde{u}_{j,v_i} = u_h^j(\mathbf{x}_{v_i})$, $\bar{u}_{v_i}^{\max}$ 和 $\bar{u}_{v_i}^{\min}$ 是共享顶点 v_i 的所有单元均值中的最大和最小值.

MLP 限制器需要所有与目标单元共享顶点的邻居单元的平均值信息, 因而需要访问多层邻居单元. 但是在非结构网格程序中, 访问多层邻居单元会增加计算时间开销, 降低计算效率, 对于三维情形尤其突出. 由于文 [19,20] 没有给出并行实现细节, 本文给出如下实现策略.

我们引入长度为 N (PHG 当前子网格节点总数) 的辅助数组 $V_{\max}^{\min}$. 该数组的每个元素存储两个浮点型变量, 用于分别记录共享该节点的所有单元均值中的最大、最小值. 之后将数组 $V_{\max}^{\min}$ 中的值赋给相关单元数据结构中, 这样就可利用 PHG 内在的子网格间通信机制, 自动实现 MPI 并行分区的信息交换. 仅需访问当前单元数据结构中存储的最大、最小值就完成限制过程. 具体实现步骤如下:

(1) 初始化辅助数组 $V_{\max}^{\min}$, 即

$$V_{v_i}^{\max} = \text{FLT_MIN}, \quad V_{v_i}^{\min} = \text{FLT_MAX}, \quad \forall v_i \in \mathcal{V}(\mathcal{T}).$$

其中 FLT_MIN 和 FLT_MAX 表示 C 语言的最小、最大浮点数.

(2) 遍历当前子网格中所有四面体单元, 获取单元中所有顶点在 PHG 子网格中的局部编号, 将单元平均值与 $V_{\max}^{\min}$ 在这些编号顶点处的最大、最小值进行比较, 根据比较结果更新 $V_{\max}^{\min}$ (见图 4).

(3) 再次遍历当前子网格中所有四面体单元, 将每个单元中各顶点对应的数组 $V_{\max}^{\min}$ 中的最大、最小值以及顶点的全局编号赋值到 PHG 数据结构的单元自由度上.

(4) 利用 PHG 通信接口获取面邻居单元顶点的最大、最小值以及这些顶点的全局编号, 将当前子网格界面单元上全局编号和接收的相同全局顶点编号处的最大、最小值进行比较, 完成当前进程界面顶点的数据更新.

(5) 利用单元数据结构提供的顶点最大、最小值信息, 用 (4.1) 式计算 MLP 限制因子, 对单元的斜率自由度进行修正.

5. 数值算例

本节给出了一些数值算例, 测试求解器 Libdgphg 的并行效率、计算精度、以及 MLP 限制器的优点. 我们取 CFL 条件数为 0.3, 应用 HLLC 数值通量 [2], 间断有限元空间采用 P^1 次多项

式 [33]. 所有计算都是在中国科学院计算机网络信息中心超级计算机“元”上进行的. “元”的每个节点有 2 个处理器, 处理器型号是 Intel Xeon E5-2680 V3 CPU(2.5GHz、12 核). 除 5.1 外, 所有数值算例中的非结构四面体网格均是由网格生成包 Gmesh 生成.

为测试程序的并行效率, 我们考虑强可扩展性并行效率 [34], 定义为

$$E = \frac{N_1 \times T_1}{N_p \times T_p}, \tag{5.1}$$

其中 T_1 是用 N_1 个核的计算时间, T_p 是用 N_p 个核的计算时间.

5.1. 三维线性标量问题

为测试采用 MLP 限制器和 Minimod 限制器的计算精度, 考虑三维线性问题 [35]:

$$u_t + 0.8u_x + 0.9u_y + u_z = 0, \quad 0 \le x, y, z \le 1.0. \tag{5.2}$$

解析解为 $u(x, y, z, t) = \sin \pi(x - 0.8t) \sin \pi(y - 0.9t) \sin \pi(z - t)$, 计算终止时间是 $T = 0.5$, 给定精确边界条件和初始条件. 采用正方体一分为六生成的规则四面体网格, 见图 5.

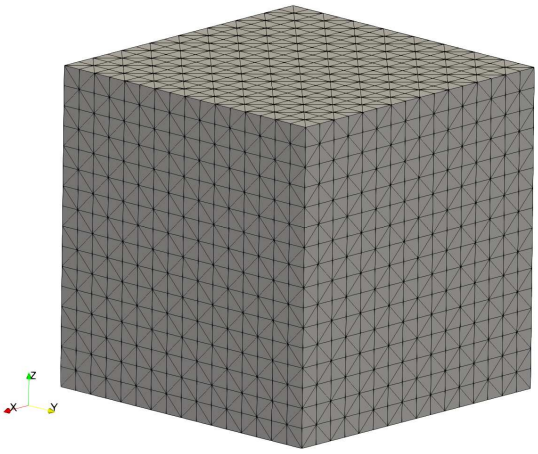

图 5　三维线性问题所用网格

表 1 显示了三维线性问题分别用 Minimod 限制器 [22] 和 MLP 限制器情况下的精度收敛情况. 我们输出计算区域正方体对角线上的 100 个采样点值, 计算它们的 $\|u - u_{\text{exact}}\|_1$ 和 $\|u - u_{\text{exact}}\|_\infty$ 收敛精度. 由表 1 可以看出, Minimod 限制器对于 $\|\cdot\|_1$ 和 $\|\cdot\|_\infty$, 出现降阶情况, 收敛精度在一阶附近. 而 MLP 限制器收敛精度达到了理论二阶精度.

5.2. 三维激波管问题

考虑圆柱管道激波管问题 [19], 初始条件如下

$$U|_{t=0} = \begin{cases} U_L & \text{if } x \le 0, \\ U_R & \text{if } x > 0 \end{cases} \tag{5.3}$$

表 1 线性问题收敛精度

Grid	Minimod Limiter				MLP Limiter			
	$\|\cdot\|_1$	EOC	$\|\cdot\|_\infty$	EOC	$\|\cdot\|_1$	EOC	$\|\cdot\|_\infty$	EOC
384	1.412E-01		4.322E-02		1.127E-01		3.749E-01	
3072	7.279E-02	0.96	2.762E-01	0.65	3.974E-02	1.50	1.206E-01	1.64
24576	3.457E-02	1.07	1.342E-01	1.04	1.070E-02	1.89	3.789E-02	1.69
196608	1.734E-02	1.00	6.718E-02	1.00	2.772E-03	1.95	1.045E-02	1.86
1572864	8.594E-03	1.01	3.352E-02	1.00	6.795E-04	2.03	2.627E-03	1.99
12582912	4.267E-03	1.04	1.605E-02	1.06	1.610E-04	2.08	6.279E-04	2.06

其中

$$U_L := \begin{bmatrix} \rho_L \\ u_L \\ v_L \\ w_L \\ p_L \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \qquad U_R := \begin{bmatrix} \rho_R \\ u_R \\ v_R \\ w_R \\ p_R \end{bmatrix} = \begin{bmatrix} 0.125 \\ 0 \\ 0 \\ 0 \\ 0.1 \end{bmatrix}.$$

计算区域为半径$r = 0.1$, 长度$l = 1.0$的三维圆柱体, 坐标原点是计算区域的中心点. 激波管内壁采用固壁边界条件, 左右两端使用对称边界条件, 计算终止时间$T = 0.2$. 激波管问题就是一维的Riemann问题, 精确解由Riemann求解器得到. 采用Gmsh软件生成非结构四面体网格如图6所示.

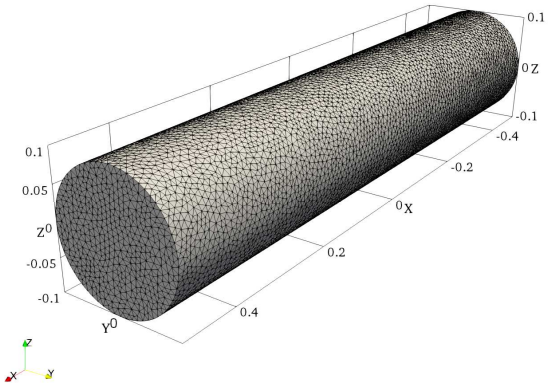

图 6 **三维激波管问题所用网格**

图7选取了圆柱体中心线$x = 0$上100个采样点值, 显示了利用Minimod限制器和MLP限制器求解三维激波管问题密度ρ, 压力p对比情形. 从图中可以看出, 两种限制器计算结果与精确解相比较, MLP限制器耗散性更小, 计算精度更准确, 效果更好.

表2显示了Libdgphg关于激波管问题的强可扩展性测试结果, 所用单元总数为1198244. 从表中可以看出随着CPU核数的增加, 总计算时间减小, 并行效率下降. 当核数为512以下时,

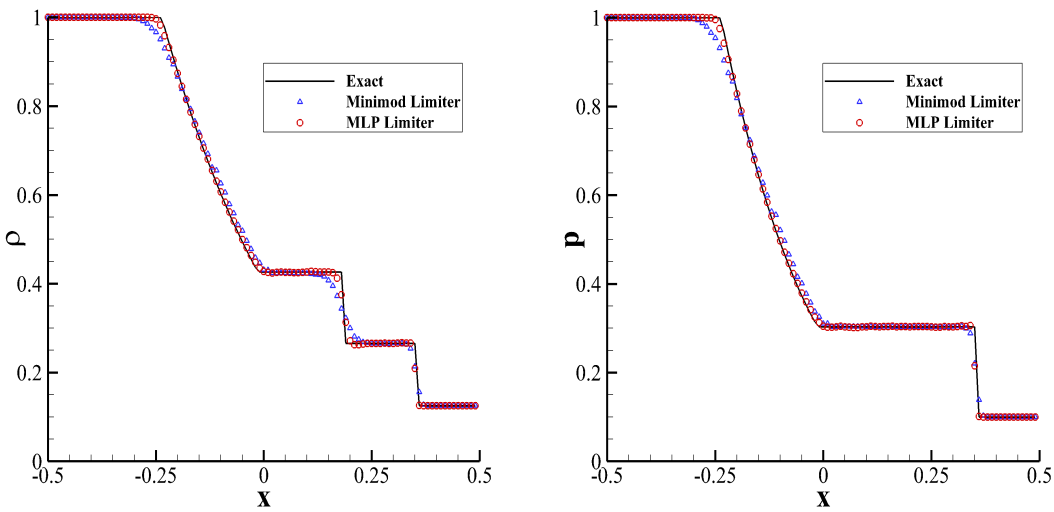

图 7　三维激波管问题用 Minimod 限制器和 MLP 限制器计算的密度 ρ, 压力 p 的对比图. 计算终止时间 $T = 0.2$, 网格单元总数 125265

并行效率在 60% 及以上. 而 1024 个核时, 并行效率下降为 50%, 这是由于计算规模限制, 分配到每个核的单元仅为 1170 个, 计算量小导致.

表 2　激波管问题强并行可扩展性测试结果, 单元总数: 1198244

N_p	Wall time (seconds)	wall time/step	Efficiency E_s	cells/proc
24	2107.0071	2.3516	1.000	49927
64	799.2690	0.8920	0.989	18772
128	415.7601	0.4640	0.950	9361
256	304.0475	0.3393	0.650	4680
512	166.8158	0.1862	0.592	2340
1024	98.7096	0.1102	0.500	1170

5.3. 三维球对称爆炸问题

考虑球对称爆炸问题 [19], 初始状态下计算区域的核心部分为高压高密度区, 外围则为低压低密度区. 设定计算区域为半径为 $R = 1.0$ 的球体, 核心部分区域半径为 $R_e = 0.4$, 计算终止时间为 $T = 0.5$, 初始条件如下

$$U|_{t=0} = \begin{cases} U_{\text{int}} & \text{if } r \le R_e, \\ U_{\text{ext}} & \text{if } r > R_e \end{cases} \tag{5.4}$$

其中

$$U_{\text{int}} := \begin{bmatrix} \rho_{\text{int}} \\ u_{\text{int}} \\ v_{\text{int}} \\ w_{\text{int}} \\ p_{\text{int}} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \qquad U_{\text{ext}} := \begin{bmatrix} \rho_{\text{ext}} \\ u_{\text{ext}} \\ v_{\text{ext}} \\ w_{\text{ext}} \\ p_{\text{ext}} \end{bmatrix} = \begin{bmatrix} 0.125 \\ 0 \\ 0 \\ 0 \\ 0.1 \end{bmatrix}.$$

采用Gmsh软件生成的非结构四面体网格见图8.

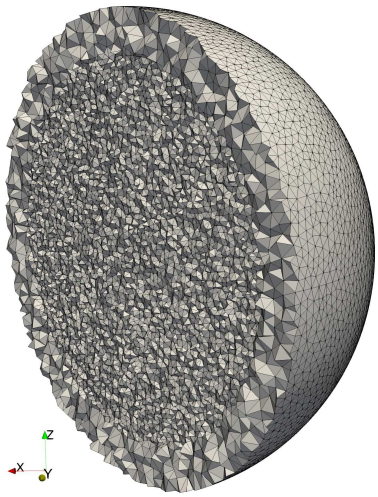

图 8 球对称爆炸问题所用网格 (为显示内部网格单元结构, 截取半个球体)

图9选取了计算球体半径上100个样本点值, 显示了分别采用Minimod限制器和MLP 限制器计算球对称爆炸问题密度ρ, 压力p及能量E对比情况. 从图中可以看出, MLP限制器相比于MiniMod限制器耗散更小, 计算效果更好.

表3 显示Libdgphg库计算球对称爆炸问题的强可扩展性并行效率, 单元总数10,691,458. 从表中可以看出, 随着计算核数增加, 计算时间减少, 并行效率下降. 但是, 千核以下并行效率始终保持在60% 以上, 说明Libdgphg库适合中大规模并行计算.

5.4. 三维激波与锥体相互作用问题

为了测试Libdghph库计算包含复杂波结构的三维流动效果, 我们考虑激波与圆锥体相互作用问题 [19,36]. 当激波冲击圆锥表面时, 会产生反射激波, 并在圆锥后端后产生三维涡流结构. 该问题计算区域包括在x方向区间$[-1.5, 3]$上的一个半圆柱, 其在$y-z$平面上的半径为$R = 2.25$. 半圆柱内部包括一个高度是1的半圆锥体, 它的顶点半径是0.02, 底面半径是0.5, 顶点在原点. 初始条件是马赫数$M_s = 1.3$ 的运动激波, 如下

$$U|_{t=0} = \begin{cases} U_L & \text{if } x < -1.0, \\ U_R & \text{if } x \geq -1.0 \end{cases} \tag{5.5}$$

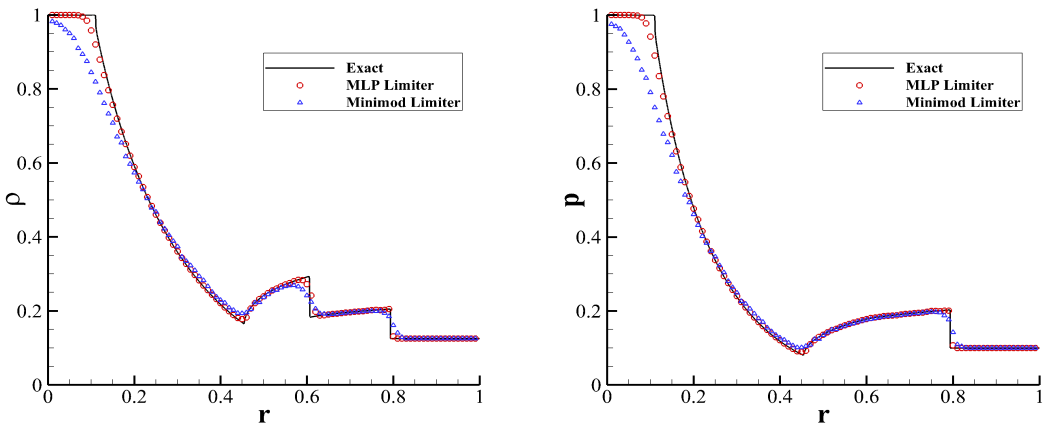

图 9　三维球对称爆炸问题分别用 Minimod 限制器和 MLP 限制器的密度 ρ 和压力 p 对比图. 计算终止时间 $T = 0.25$, 网格单元总数 10691458

表 3　球对称爆炸问题强并行可扩展性测试结果, 单元总数: 10,691,458

N_{proc}	Wall time (seconds)	wall time/step	Efficiency E_s	cells/proc
24	45103.2156	21.8735	1.000	445478
64	17110.7469	8.2981	0.989	167054
128	8820.2380	4.2775	0.959	83528
256	4544.2773	2.2028	0.931	41764
512	3206.0347	1.5541	0.660	20882
1024	1643.8064	0.7968	0.643	10441

其中

$$U_L := \begin{bmatrix} \rho_L \\ u_L \\ v_L \\ w_L \\ p_L \end{bmatrix} = \begin{bmatrix} 2.122 \\ 0.442 \\ 0 \\ 0 \\ 1.805 \end{bmatrix}, \qquad U_R := \begin{bmatrix} \rho_R \\ u_R \\ v_R \\ w_R \\ p_R \end{bmatrix} = \begin{bmatrix} 1.4 \\ 0 \\ 0 \\ 0 \\ 1.0 \end{bmatrix}.$$

采用 Gmsh 软件生成非结构四面体网格见图 10.

图 11 显示了该问题分别用 Minimod 限制器和 MLP 限制器计算的三维密度云图的对比, 计算终止时间分别为 2.0 和 2.5. 从图中计算结果可以看出, MLP 限制器的等值线锐化程度明显, 因此相比于 Minimod 限制器, MLP 限制器能够捕捉更高分辨的多维激波结构.

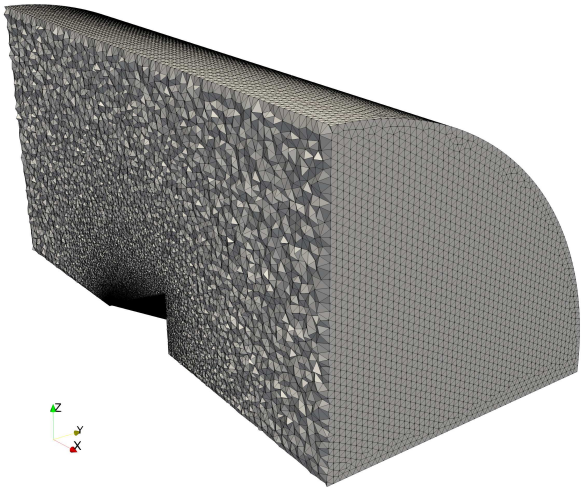

图 10 激波与圆锥体相互作用问题所用网格 (为显示内部网格单元结构, 截取半个圆柱体)

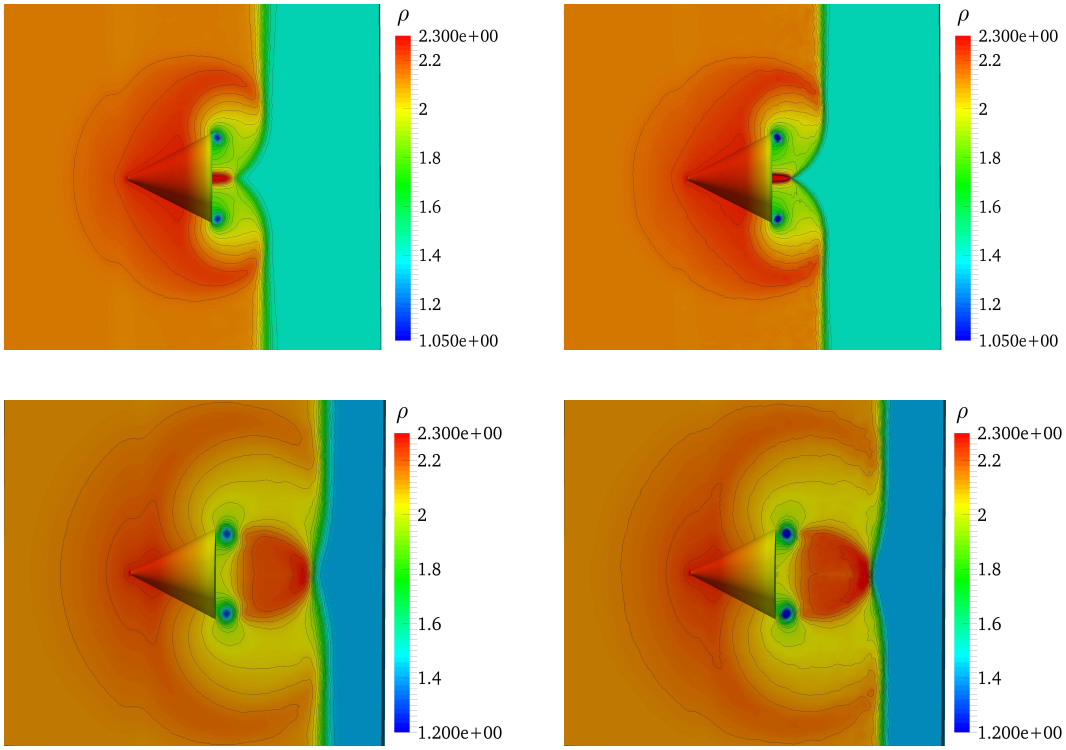

图 11 激波与圆锥体相互作用问题使用Minimod 限制器和MLP 限制器的密度ρ云图侧视图. 左列: Minimod 限制器, 右列: MLP 限制器; 上面: 计算终止时间$T = 2.0$, 下面: 终止时间 $T = 2.5$. 网格单元总数5,868,572

6. 结　论

本文简述了基于PHG平台开发的在三维四面体网格上用间断有限元方法求解欧拉方程的并行求解器Libdgphg, 实现了用模态基 P^1 元、多种数值通量的数值求解. 采用具有多维分辨特性的MLP限制器. 为缓解MLP限制器模板过大的问题, 通过引入辅助数组和遍历单元来收集共顶点邻居单元均值的最大、最小值的策略, 避免限制时重复访问共顶点邻居单元, 并将这些最大、最小值赋予PHG的单元自由度, 满足了PHG子网格之间通信接口的要求. 数值算例验证了MLP限制器比传统的MUSCL限制器有更高的分辨率, Libdgphg解法器有良好的并行可扩展性能, 在千万单元、千核测试中具有60%以上的并行效率, 可用于求解大规模问题. 下一步, 我们将开展将Libdgphg库扩展到高阶精度间断有限元, 隐式时间求解, Navier-Stokes方程等工作.

致谢. 本文中的计算是在中国科学院科学与工程计算国家重点实验室高性能计算机系统, 以及中国科学院计算机网络信息中心超级计算机上完成的.

参 考 文 献

[1] Anderson J D, 姚朝辉, 周强 (编译). 计算流体力学入门 [M]. 清华大学出版社, 北京, 2010.

[2] Toro E F. Riemann Solvers and Numerical Methods for Fluid Dynamics[M]. Springer-Verlag, Berlin Heidelberg, 2009.

[3] Blazek J. Computational Fluid Dynamics: Principles and Applications[M]. Elsevier Science, Oxford, 2005.

[4] Shu C W. Discontinuous Galerkin methods: General approach and stability[J]. Numerical Solutions of Partial Differential Equations, Adv. Courses Math. CRM Barcelona, 2009, 149–201.

[5] Rivière Béatrice. Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation[M]. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2008.

[6] Cheng J and Shu C W. High order schemes for CFD: A Review[J]. 计算物理, 2009, 26(5): 633–655.

[7] Cockburn B, Shu C W. Runge-Kutta discontinuous Galerkin Methods for convection-dominated problems[J]. Journal of Scientific Computing, 2001, 16(3): 173–259.

[8] Cockburn B, Shu C W. The Runge-Kutta discontinuous Galerkin method for conservation Laws V: Multidimensional systems[J]. Journal of Computational Physics, 1998, 141(2): 199–224.

[9] 刘岩, 关振群, 张洪武, 张占群. 面向大规模科学计算的三维 Delaunay 快速插点算法 [J]. 中国科学: 物理学力学天文学, 2012, 42(2): 192–198.

[10] Peraire J, Vahdati M, Morgan K and Zienkiewicz O C. Adaptive remeshing for compressible flow computations[J]. Journal of Computational Physics, 1987, 72(2): 449–466.

[11] Naddeia F, Plataa M L, Couailliera V and Coquel F. A comparison of refinement indicators for p-adaptive simulations of steady and unsteady flows using discontinuous Galerkin methods[J]. Journal of Computational Physics, 2019, 376: 508–533.

[12] 杨雨薇, 虞佳磊, 张亚萍, 李幸刚, 邱晓朴. 基于非结构自适应网格的二维 Euler 方程数值求解方法研究 [J]. 应用数学和力学, 2016, 37(9): 981–992.

[13] Yu X J, Wu D, Xu Y. Three dimensional discontinuous Galerkin methods for Euler equations on adaptive conforming meshes[J]. Comput. Phys. Commu., 2011, 182: 1771–1775.

[14] Conroya C J, Kubatko E J, Nappi A, et al.. HP discontinuous Galerkin methods for parametric, wind-driven water wave models[J]. Advances in Water Resources, 2018, 119: 70–83.

[15] 夏轶栋, 伍贻兆, 吕宏强, 宋江勇. 高阶间断有限元法的并行计算研究 [J]. 空气动力学学报, 2011, 29(5): 537–541.

[16] Fujii K. Progress and future prospects of CFD in aerospace-wind tunnel and beyond[J]. Progress in Aerospace Sciences, 2005, 41(6): 455–470.

[17] Kroll N. ADIGMA-A European project on the development of adaptive higher-order variational methods for aerospace applications[P]. 47th AIAA Aerospace Sciences Meeting, 2009.

[18] Slotnick J, Khodadoust A, Alonso J, Darmofal D, et al.. CFD vision 2030 Study: A path to revolutionary computational aerosciences[P]. NASA Langley Research Center, 2013, NASA/CR-2014-218178.

[19] Park J S, Kim C. Multi-dimensional limiting process for finite volume methods on unstructured grids[J]. Computers & Fluids, 2012, 65: 8–24.

[20] Park J S, Kim C. Higher-order multi-dimensional limiting strategy for discontinuous Galerkin methods in compressible inviscid and viscous flows[J]. Computers & Fluids, 2014, 96: 377–396.

[21] Zhang F, Liu J, Chen B S. Modified multi-dimensional limiting process with enhanced shock stability on unstructured grids[J]. Computers & Fluids, 2018, 161: 171–188.

[22] Barth T J, Jespersen D C. The design and application of upwind schemes on unstructured meshes[P]. 27th AIAA aerospace sciences meeting, Reno RV, 1989, No. AIAA 89–0366.

[23] Touze C Le, Murrone A, and Guillard H. Multislope MUSCL method for general unstructured meshes[J]. Journal of Computational Physics, 2015, 284: 389–418.

[24] Sohn S I. A new TVD-MUSCL scheme for hyperbolic conservation laws[J]. Computers & Mathematics with Applications, 2005, 50: 231–248.

[25] Panagiotis T. Extended bounds limiter for high-order finite-volume schemes on unstructured meshes[J]. Journal of Computational Physics, 2018, 362: 69–94.

[26] Shu C W . TVB uniformly high-order schemes for conservation laws[J]. Mathematics of Computation, 1987, 49(179): 105–121.

[27] 张林波, 郑伟英, 卢本卓等. 并行自适应有限元软件平台PHG 及其应用 [J]. 中国科学: 信息科学, 2016, 46(10): 1442–1464.

[28] Zhang L B. A parallel algorithm for adaptive local refinement of tetrahedral meshes using bisection[J]. Numerical Mathematics: Theory, Methods and Applications, 2009, 2(1): 65–89.

[29] 刘辉, 冷伟, 崔涛. 并行自适应有限元计算中的负载平衡研究 [J]. 数值计算与计算机应用, 2015, 36(3): 166–184.

[30] 刘辉, 张林波. 自适应有限元常用标记策略及其在 PHG 中的并行实现 [J]. 数值计算与计算机应用, 2009, 30(4): 315–320.

[31] Shu C W and Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes[J]. Journal of Computational Physics, 1988, 77(2): 439–471.

[32] Spiteri R J, Ruuth S J. A new class of optimal high-order strong-stability-preserving time discretization methods[J]. Siam Journal on Numerical Analysis, 2003, 40(2): 469–491.

[33] Luo H, Baum J D and Löhner R. A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids[J]. Journal of Computational Physics, 2008, 227: 8875–8893.

[34] 王承尧, 王正华, 杨晓辉. 计算流体力学及其并行算法 [M]. 国防科技大学出版社, 长沙, 1999.

[35] Zhou T, Li Y F and Shu C W. Numerical comparison of WENO finite volume and Runge-Kutta discontinuous Galerkin methods[J]. J. Sci. Comput., 2001, 16: 145–171.

[36] Dumbser M, Käser M, Titarev V A and Toro E F. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems[J]. Journal of Computational Physics, 2007, 226: 204–243.