

## 非结构网格的并行多重网格解算器\*

李宗哲, 王正华, 姚路, 曹维

(国防科学技术大学 计算机学院 软件研究所, 湖南 长沙 410073)

通讯作者: 李宗哲, E-mail: lzz144@163.com, http://www.nudt.edu.cn

**摘要:** 多重网格方法作为非结构网格的高效解算器,其串行与并行实现在时空上都具有优良特性.以控制方程离散过程为切入点,说明非结构网格在并行数值模拟的流程,指出多重网格方法主要用于求解时间推进格式产生的大规模代数系统方程,简述了算法实现的基本结构,分析了其高效性原理;其次,综述性地概括了几何多重网格与代数多种网格研究动态,并对其并行化的热点问题进行了重点论述.同时,针对非结构网格的实际应用,总结了多重网格解算器采用的光滑算子;随后列举了非结构网格应用的部分开源项目软件,并简要说明了其应用功能;最后,指出并行多重网格解算器在非结构网格应用中的若干关键问题和未来的研究方向.

**关键词:** 非结构网格;computational fluid dynamics(CFD);并行计算;多重网格;高效解算器

中图法分类号: TP316 文献标识码: A

中文引用格式: 李宗哲,王正华,姚路,曹维.非结构网格的并行多重网格解算器.软件学报,2013,24(2):391-404. <http://www.jos.org.cn/1000-9825/4241.htm>

英文引用格式: Li ZZ, Wang ZH, Yao L, Cao W. Parallel multigrid solver for unstructured grid. Ruanjian Xuebao/Journal of Software, 2013,24(2):391-404 (in Chinese). <http://www.jos.org.cn/1000-9825/4241.htm>

### Parallel Multigrid Solver for Unstructured Grid

LI Zong-Zhe, WANG Zheng-Hua, YAO Lu, CAO Wei

(Institute of Software, College of Computer, National University of Defense Technology, Changsha 410073, China)

Corresponding author: LI Zong-Zhe, E-mail: lzz144@163.com, http://www.nudt.edu.cn

**Abstract:** As an unstructured-grid high efficient solver, the multigrid algorithm, with its serial and parallel application, can achieve the optimal properties of being on time and having space complexity. To illustrate the numerical simulation process of an unstructured grid, this paper begins with the discretization of governing equations and points out that the multigrid algorithm is mainly used for solving large scale algebraic equation, which is derived from the time marching scheme. For the multigrid algorithm, the study briefly describes its basic structure and efficient principle. Secondly, the paper reviews research that trend about the geometric multigrid and algebraic multigrid and discusses the basic design principles and hot topics on parallelization. At the same time, for the practical application of unstructured grid, the paper summarizes and classifies many smoothers, followed by examples of open source software about unstructured grid industrial application. Finally, some applications and key problems in this field are highlighted, as well as the future progress of parallel multigrid solver on unstructured grid.

**Key words:** unstructured grid; computational fluid dynamics (CFD); parallel computing; multigrid; efficient solver

非结构网格作为计算流体力学(computational fluid dynamics,简称 CFD)的一个重要研究分支,对复杂几何外形具有较强的适应性,弥补了结构网格不能在任意形状、任意连通区域内进行网格剖分的缺欠,因而被 CFD 工程计算广泛采用.与结构网格相比,非结构网格可以使用更加灵活的几何形体.但是,非结构网格的存储格式比较复杂,不仅需要表示节点的坐标,而且需要标识单元的邻接关系.从数值计算角度来看,几何形体与空间离

\* 基金项目: 国家重点基础研究发展计划(973)(2009CB723803)

收稿时间: 2011-02-23; 定稿时间: 2012-04-20

散精度密切相关,而网格的数据集存储格式又影响着计算效率<sup>[1]</sup>.

一直以来,高精度格式与高效率的数值方法构成了非结构网格研究的两个热点:在高精度格式方面,离散 Galerkin<sup>[2]</sup>、谱体积<sup>[3]</sup>、谱差分<sup>[4]</sup>等方法的引入,使得非结构网格在计算格式上具有了二阶以上的计算精度,维持了较强的数值稳定性,在一定程度上满足了数值模拟高分辨率的需求;在高效率算法方面,由于 CFD 工程应用的瓶颈往往出现在求解大规模的代数系统方程上,可以说在 CFD 建立之初,实现更加快速、有效的求解方法,成为广大 CFD 研究者不断追求的目标,因此,经过几十年的发展,各种高效的数值求解方法仍在不断地出现和完善,其中,多重网格方法<sup>[5-7]</sup>的应用和改进就是这一领域比较活跃的代表,由于其串行的时间和空间复杂度均为  $O(n)$ ,满足渐进最优的性质,适用于线性和非线性系统方程的求解,且不受控制方程时间和空间离散格式的限制,如显式格式、隐式格式都可以获得较快的收敛速度,因而,非结构网格的高效解算器中必然涉及多重网格方法.

同时,非结构网格的 CFD 计算问题一般比较庞大,某些问题甚至是串行计算机无法解决的,因而需要大规模并行计算平台的支持.并且,解算器的并行实现可以大量缩短计算时间上的开销,更加真实地模拟和再现物理环境,在很大程度上满足了高精度、多目标、实时性方面的需求.另一方面,非结构网格的 CFD 软件开发却远远没有跟上计算机硬件发展的速度<sup>[8]</sup>,且并行扩展能力不高,规模较大的计算问题往往使用较少的计算节点,使得并行计算平台的优越性得不到有效的发挥.另外,随着计算机体系结构的发展,使用 CPU+GPU 的计算模式<sup>[9,10]</sup>越来越多地被 CFD 领域所采用.可以看出,非结构网格上的高效解算器必须具有良好的并行可扩展性,适用于不同的硬件环境和计算平台.因而,非结构网格中多重网格的并行化应用研究,也就成为近年来 CFD 数值解算器研究的热点问题.

本文主要以多重网格求解非结构网格下大规模系统方程为主线,详细分析了其应用研究过程中的技术难点和热点.首先,从非结构网格 CFD 的控制方程出发,区别对待空间离散和时间推进格式,引出工程应用中的计算需求以及相应高效求解方法;其次,对多重网格方法的起源、结构和求解原理进行简要的说明,重点阐述几何多重网格和代数多重网格各自研究中的技术热点,归纳其串、并行计算的层次构建方法,并对几何多重网格方法和代数多重网格方法进行比较,同时总结了实际应用中光滑算子的选取策略;最后,简要叙述部分相关开源项目,同时对未来研究方向做进一步的展望.

## 1 控制方程离散化

非结构网格下的 CFD 数值模拟问题源于对 Navier-Stokes 方程或 Euler 方程的离散,以 3D 非定常可压 NS 方程为例,其守恒形式可以表示为

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = S \quad (1)$$

其中,  $Q$  表示解向量,  $F, G, H$  分别表示  $x, y, z$  方向上的通量项,  $S$  代表有源项(当体积力和体积热流可以忽略时等于 0). 方程(1)的通量项离散化方法主要包括有限差分、有限元、有限体积、谱方法等,根据非结构网格存储格式的不同,又可以分为单元中心型(cell-center)和节点中心型(node-center). 关于这两种形体格式孰优孰劣的讨论,长期以来一直没有定论. 单元中心型的优势在于流场特性的分辨率较高,但计算量大;节点中心型的优势在于计算量小,对网格质量不太敏感. 文献[11,12]中对比了 6 种单元中心格式和两种节点中心格式,从理论分析和数值实验的角度阐明:单元中心最近邻接格式(cell-centered nearest-neighbor,简称 CC-NN)适用于无粘通量离散,节点中心格式(node-centered,简称 NC)适用于粘性通量离散,各自具有较优的时空复杂度.

对于空间离散化过程,以有限体积方法为例,方程(1)最终可以写成半离散化形式:

$$\frac{\partial Q}{\partial t} \Delta V_i = R(Q_i, Q_j) \quad (2)$$

其中,  $\Delta V_i$  表示控制体  $i$  的体积,  $R$  为空间离散格式形成的右端项,  $Q_i$  为控制体  $i$  的解向量平均值. 需要说明的是,其他空间离散方法也可以得到类似于方程(2)的半离散化形式.

至此,空间离散后转化为时间常微分方程的积分过程,即方程(2)的求解. 在时间推进方法上,存在显示和隐

式两种方法.比较而言,显示方法的单步计算速度快,内存需求小,数值稳定性强烈依赖于 CFL(Courant-Friedrichs-Lewy)数.早期的非结构显格式解算器不仅计算方式简单,时空复杂度也比较高,而且数值稳定性较差.自从 Jameson 引入多步 Runge-Kutta 方法<sup>[13]</sup>后,实现了高阶时间精度,随后被广泛用于 CFD 商业软件. $m$  步 Runge-Kutta 方法可表述为

$$\begin{cases} Q^{(0)} = Q^n \\ Q^{(m)} = Q^{(0)} + \alpha_m \cdot \frac{\Delta t}{\Delta V_i} \cdot R(Q^{(m-1)}) \\ Q^{n+1} = Q^{(m)} \end{cases} \quad (3)$$

其中, $Q^n$  和  $Q^{n+1}$  分别表示第  $n$  层和  $n+1$  层上的值,  $m$  表示步数, $\alpha_m$  为第  $m$  项系数.Jameson 在文献<sup>[14]</sup>中使用了多重网格化的 Runge-Kutta 方法加速 Euler 方程的收敛过程,取得了较好的效果.

另一类经典的显式方法是总变差递减(total variation diminishing,简称 TVD)型的 Runge-Kutta 方法<sup>[15]</sup>,具有较强的数值稳定性,其系数 $\alpha_m$  相对不固定,因而存在多种极大化 CFL 参数的优化版本.其他显式格式方法的相关细节可参见文献<sup>[16]</sup>.

相比而言,隐式方法单步计算的时间步长降低了对 CFL 数的限制,某些问题的离散化甚至是无条件收敛的,因而数值稳定性更高,引入的问题就是必须求解大型稀疏线性方程组.将方程(2)采用一阶向后 Euler 差分,省略中间步骤,可以简化为

$$\left( \frac{\Delta V_i}{\Delta t} \cdot I - \frac{\partial R^n}{\partial Q^n} \right) \cdot \Delta Q^{n+1} = R(Q_i^n, Q_j^n) \quad (4)$$

其中, $\Delta Q^{n+1} = Q^{n+1} - Q^n$ .上式等价于求解  $Ax=b$  的原型问题.非结构网格与结构网格不同,不能直观地得到矩阵  $A$  的性质,如对角占优、对称正定、三对角型、五对角型等,从而其数值解算器不能完全照搬结构网格.例如,1D 结构网格并行计算使用的 Wang 分裂法<sup>[17]</sup>(又称追赶法),只适用于三对角型矩阵的情况.

线性系统方程的数值求解,大体可以分为直接法和迭代法两大类.直接法以矩阵分裂为基础,易于实现,理论上可以求得精确解,但计算量、存储量较大,适用于对角型的特殊矩阵;迭代法只提供近似解,计算量存储量较小,易于并行实现.所以,非结构网格的解算器,尤其是并行解算器,在满足相应精度格式的前提下,很少采用直接法,而是采用迭代法求解.迭代法又可分为 3 类:以矩阵分裂为基础的经典迭代法、以 Krylov 子空间为主体的投影方法、层次化的多重网格方法.Krylov 子空间迭代方法通常加入预条件技术,以取得更好的收敛效果<sup>[18-20]</sup>.有关迭代法的具体细节可参见 Saad 的专著<sup>[21]</sup>和 Duff 的综述<sup>[22]</sup>.多重网格方法,无论是经典迭代方法还是 Krylov 子空间投影方法,都可以作为多重网格方法的光滑算子,达到消除高频误差的效果.

## 2 多重网格解算器

多重网格方法最早产生于 20 世纪 60 年代,Fedorenko 和 Bakhvalov 开发了两层网格的插值校正算法用于求解泊松型偏微分方程.之后,层次化误差校正的高效性逐渐被人们所认识,且越来越多地应用到数值计算领域.其误差校正的原理是:将高频误差采用光滑算子(如经典迭代方法)消除,低频误差在网格由细层转化为粗层的过程中,相应地转化为粗网格层上的高频误差,同样采用光滑算子进行消除,最后,将计算结果插值到细层网格,修正先前结果,从而达到快速收敛的目的.

对于线性方程组的原型问题  $Ax=b$ ,定义网格层次,设定  $A^1=A$ ,网格空间层次序列  $\Omega^1 \supset \Omega^2 \supset \dots \supset \Omega^m$ .从空间  $\Omega^{i+1}$  向空间  $\Omega^i$  的插值算子定义为  $P_{i+1}^i$ ,空间  $\Omega^i$  到空间  $\Omega^{i+1}$  的限制算子定义为  $R_i^{i+1}$ ,空间  $\Omega^i$  的光滑算子为  $S^i$ ,则空间  $\Omega^{i+1}$  上的矩阵算子  $A^{i+1} = R_i^{i+1} A^i P_{i+1}^i$ .具体的多重网格可见算法 1.

**算法 1.**  $MG(A^k, x^k, b^k)$ .

if  $k=m$  then

    精确计算  $A^k x^k = b^k$

else

```

前光滑,迭代  $\nu_1$  次  $x^k = S^k(A^k, x^k, b^k)$ 
粗网格校正:
    残差限制,  $b^{k+1} = R_k^{k+1}(b^k - A^k x^k)$ 
    设定初值,  $x^{k+1} = 0$ 
     $k+1$  次递归, for  $j=1$  to  $\gamma$  do  $x^{k+1} = MG(A^{k+1}, x^{k+1}, b^{k+1})$ 
    误差校正,  $x^k = x^k + P_{k+1}^k x^{k+1}$ 
后光滑,迭代  $\nu_2$  次  $x^k = S^k(A^k, x^k, b^k)$ 
endif

```

当  $\gamma=1$  时,该算法的迭代过程被称为 V 循环;当  $\gamma=2$  时,该算法迭代过程被称为 W 循环.如图 1 所示,图中原点表示光滑过程,实线表示传递算子(插值、限制算子).多重网格方法不断改进过程中,在这两种循环方式的基础上,发展出了多种循环结构,如 N 型循环、FMG(full multigrid)、K 型循环等结构.其中,FMG,W,V 循环应用最为广泛.串行计算时,这 3 种循环方式的空间和时间复杂度均为  $O(n)$ .从大量数值实验结果来看,其性能相差并不明显,相对而言,FMG 要好于 W 循环和 V 循环,W 循环又好用 V 循环.另一方面,并行计算环境下,FMG,W,V 循环相应的复杂度分别为  $O(\log^2 n)$ ,  $O(\sqrt{n})$ ,  $O(\log n)$ .可见,V 循环在并行计算时间上具有较快的收敛特性.

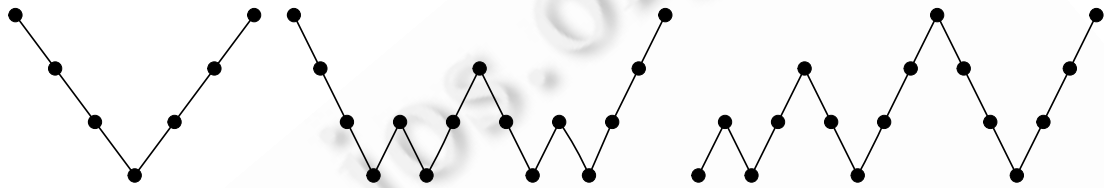


Fig.1 V-Cycle, W-cycle and FMG-cycle

图 1 V 循环、W 循环和 FMG 循环

为了使得多重网格方法达到理想的性能要求,其设计和应用过程中最关键的两个问题是网格层次结构的构建和光滑算子的选取.网格层次根据几何成网格和代数矩阵的不同,其构建过程也相互独立.无论是几何形式还是代数形式的插值、限制算子,其功能都是用于构建网格层次和各层间的数值传递,因而可以归结为第 1 个问题的组成部分.而光滑算子的优劣,将严重影响各层上高频误差的消除,决定了收敛速度的快慢;同时,非结构网格计算环境下的多重网格光滑算子在加速收敛、处理各向异性问题等方面又存在其特殊性.因此,下面将分别从几何多重网格、代数多重网格的并行化层次构建、光滑算子选取策略加以说明.

## 2.1 几何多重网格

早期的多重网格方法是基于结构网格的几何形体层次结构,因而得名为几何多重网格(geometry multigrid, 简称 GMG).这种层次结构的构建,由于能够较快地收敛到稳定状态,很快被 CFD 计算所采用.最初的应用主要采用显式时间格式和几何多重网格相结合的方法加速其收敛过程<sup>[14]</sup>.之后,根据不同物理问题的计算特性以及非结构网格的具体需求,对几何多重网格的应用和改进<sup>[17,23-25]</sup>层出不穷,体现出几何多重网格在数值计算上强大的优越性.其中的关键问题是构建几何网格的层次结构,即网格粗化,这也是其应用中最受关注的问题.

### 2.1.1 GMG 串行粗化

在网格层次构建的串行算法中,结构网格上,GMG 层次构建比较简单,可以直接通过消除相应坐标方向上的网格线得到更粗的网格.相比之下,非结构网格就显得比较复杂,大体上可以分为 4 种类型:

- (1) 从粗网格递归的构建细网格<sup>[26]</sup>,多采用自适应改进策略.优点是建立层次结构算法简单,且容易编程实现,细网格质量得到一定程度的修正.缺点是 3D 的网格构建方法不太成熟,且细网格质量严重依赖于粗网格层,数值稳定性与求解精度不高,因此,在大规模的 CFD 工业应用中使用较少.
- (2) 非递归地从细网格层独立构建粗网格层<sup>[27,28]</sup>,采用相互独立的方法生成粗网格,各层节点没有相互

包含关系.优点是可以根据不同问题量身定制不同的粗网格层次,因而具有较优良的性能.但是,网格层次无法自动化地生成,网格层数较少,插值和误差校正过程比较复杂,多用于节点中心型计算格式.

- (3) 基于边消除(edge-collapsing)<sup>[29,30]</sup>策略的网格粗化,先消除原有网格边及其连接点,再对剩余节点重新网格化的构建下层网格.优点是容易控制粗化过程,具有合适的计算复杂度,且自动化程度比较高.缺点是只能应用于节点中心型计算格式,多层次的粗化将改变实际物面外型,对计算精度有一定的影响,且复杂外型的适应能力较差.
- (4) 通过结块(agglomeration)<sup>[31]</sup>递归地构建粗网格层次,相互邻近的单元按照一定规则聚合成一个单元,以形成下一网格层次.这种方法适用于各种复杂的几何外形,数值收敛速度较快,且实现简单易于编程,非常适合于并行应用,也是目前应用最为广泛的方法,被 Fluent,NSU3D,USM3D 等商业软件采用.此外,结块过程中聚合单元的选取具有一定的随机性,生成的粗化层网格质量也表现出一定的不确定性,通常采用启发式规则对其聚合过程进行改良,使其具有更好的形体适应能力.另外,结块式方案还可以与代数多重网格协同开发,发挥更加优良的特性.

### 2.1.2 GMG 并行粗化

GMG 的并行化方法涉及网格分区、各区网格层次的构建、边界数值通信等.

(1) 实现有效的网格分区,依赖于非结构网格的类型以及并行系统的体系结构.图划分的问题本身是 NP-完全问题,常使用各种启发式规则.典型的图划分策略有贪婪算法、KL 算法<sup>[32]</sup>、递归坐标二分、递归图二分等.对于复杂的几何外形,如多段翼、多孔介质,上述划分容易导致子区域不连通,递归谱二分<sup>[33]</sup>法可以有效弥补上述方法的缺陷,缺点就是引入大量计算开销.多层分割技术不仅快速有效而且健壮稳定,被图分割软件广泛采用,如 Metis<sup>[34]</sup>,ParMetis<sup>[35]</sup>等.另外,多层分割还可以减小通信和维持平衡负载,一个典型的应用<sup>[36]</sup>是 Mavriplis 使用 Metis 划分飞行器外形的非结构网格(72M 节点,315M 单元),用于 2 008 个处理器的并行计算,取得接近线性的加速扩展效果.

(2) GMG 网格层次的并行构建,使用非递归方式构建网格层次的方法,自动化程度不高,需要借助于网格生成软件,尤其是网格节点规模较大时,手动生成网格层次就变得不现实.所以,并行化的网格层次构建往往采用结块(agglomeration)方式.开源软件 MGridGen/ParMGridGen<sup>[37]</sup>提供了这样的功能,不仅可以定制结块大小,而且可以控制生成网格的质量,其中,MGridGen 为串行版本,ParMGridGen 为并行版本.

在并行计算时,按照网格划分与层次构建的顺序,存在两种方式:一种是网格输入时统一建立网格层次,然后分区到各个处理器上去;另一种是先划分后并行建立网格层次.在前一种方式下,各分区细网格层将继承粗网格层的通信模式,容易导致负载不均衡;后一种方式避免了负载平衡的问题,但初始化过程和通信模式比较复杂.在现实工程应用中,可以根据不同的需求采用不同的方式,更多有关细节可参见文献[38].

(3) 通常,GMG 的并行采用完全通信模式,每个网格层次上进行一次边界数据同步.虽然在数值计算的精度上得到了保证,却大量增加了通信开销,降低了并行加速比,不利于并行可扩展性开发,严重影响 GMG 算法的高效性.为了减少并行计算时的数据交换,增大计算量与通信量的比值,往往采用计算换通信的方式.对于 GMG 而言,插值和限制的过程并不需要数据更新,而只在分区产生一次完整的 GMG 循环之后进行数据通信.更有甚者,多次进行完整的并行 GMG 循环才进行一次边界数据同步<sup>[39]</sup>,其优点是可以有效地减少通信量,带来的问题是数值精度不高,收敛速度较慢.因而无论是完整的通信模式,还是计算换通信的模式,都涉及收敛速度与并行效率的矛盾.如何有效地处理这对矛盾,仍然是今后一段时间内并行计算的数值理论分析与实践应用必须面对的问题.关于更加详细的并行通信模式可参见文献[27,40].

值得一提的是,借助高精度计算格式的发展,GMG 的非结构网格解算器在数值稳定性、高效收敛性、并行可扩展性的开发上,仍然具有较强的吸引力,与并行 GMG 网格层次化构建一起组成了当前 GMG 研究的两大热点,其中最具有代表性的是  $p$ -Multigrid<sup>[24,41]</sup>和  $hp$ -Multigrid<sup>[40]</sup>格式,使得 GMG 解算器在增加数值精度的同时,收敛速度也得到了较大的提高,在 Euler 方程和 NS 方程的并行求解上取得近似线性的加速比,有效地弥补了 GMG 在并行可扩展性方面的不足.

## 2.2 代数多重网格

与几何多重网格方法不同,代数多重网格(algebraic multigrid,简称 AMG)以系数矩阵为研究对象,不需要存储网格层次的几何信息,也不需要事先确定网格层次结构,对各向异性网格以及无网格问题都具有较强的适应能力,AMG 插值、限制算子相对比较灵活,只需要存储对应的系数矩阵,随着问题规模的扩大,可扩展性能较好,不受网格维度的限制;另一方面,在每次 AMG 迭代过程中都需要计算相应的粗化和插值矩阵,对 AMG 的高效性有一定程度的影响,因而也是 AMG 研究的一个热点问题.

### 2.2.1 AMG 串行粗化

AMG 着眼于  $Ax=b$  的原型问题.早期的研究侧重于矩阵粗化过程,因为粗化方式决定了插值、限制算子,插值、限制算子用于构建下一层的原型问题,影响网格层次间误差的传递和修正;同时,随着 AMG 网格层次的不断加深,从理论上讲,对误差的消除越来越迅速.但是单步 AMG 循环的计算时间就越长,用于矩阵结构的存储也就越大,这些因素最终将影响到 AMG 高效性.因此,特定问题确定合理的网格层数,在串行 AMG 和并行 AMG 上都具有重要的意义.遗憾的是,多数实际应用中,网格层数还是通过试探性或经验性的方法来确定,而关于其数值理论的分析更是少之又少.

AMG 串行粗化方法大体上可以分为两类:

- (1) 经典的粗化<sup>[42]</sup>方式,完全将网格节点分割为粗网格点集  $C$  和细网格点集  $F$ ,粗网格节点集  $C$  用于构建下一层的整体网格节点,递推地生成多个层次.在误差校正阶段,细网格点集  $F$  的值,将由粗网格点集  $C$  中相应的点插值得到.为了减小计算量,通常采用距离为 1 的插值策略.为了平衡数值精度与计算速度的关系,其改进策略中也有条件地加入距离为 2 的数值插值,同样能够获得快速的收敛效果.
- (2) 聚合型(aggregation)的粗化<sup>[43]</sup>方式,满足强耦合条件的多个节点聚合形成一个新的节点,所有新形成的节点构成下一层的网格节点集.这种方式非常类似于 GMG 中的结块方式,在一定程度上,两种方式是等价的.唯一不同的是,其对象一个是几何形体结构,另一个是矩阵行列.

### 2.2.2 AMG 并行粗化

基于上述两类方式的并行化,也是近年来的研究热点,相对应的并行化方法也可以分成两大类:

#### (1) 经典粗化方式的并行

主要有:基于串行版本的并行化方法,包括降耦合粗化格式  $RS0$ <sup>[42,44]</sup>、耦合粗化格式  $RS3$ <sup>[42]</sup>;基于完全并行化的方法,包括  $CLJP$ <sup>[42]</sup>和  $PMIS$ <sup>[45,46]</sup>、子区域块格式<sup>[46]</sup>;以及组合型格式  $HMIS$ <sup>[45]</sup>等.著名的开源软件包  $Hypre$ <sup>[47]</sup>提供了上述各种方法的 C++版本的实现,为用户进行二次开发提供了方便.另外,从上述各种并行化策略的优劣性来说,文献[44]的数值实验提供了一定的参照,完全并行化的方法( $CLJP,PMIS$ )以及各种组合方法具有较强适应能力,其计算性能相对较好,表现出渐进收敛因子和算法复杂度相对较低的特点.

#### (2) 聚合型粗化方式的并行化

主要包括基于光滑聚合(smoothed aggregation)<sup>[48]</sup>方式的耦合并行、降耦合并行、并行极大独立集方法以及基于非光滑聚合的双点对聚合(double pairwise aggregation)<sup>[49]</sup>方法.采用光滑聚合方式,利用对插值算子进行光滑改进收敛性,可以有效地增强聚合型 AMG 方法的数值稳定性,降低对耦合节点的依赖,提高并行扩展性能.缺点是光滑过程增大了 AMG 的启动时间,对应的开源软件包有  $ML$ <sup>[50]</sup>;而双点对聚合方法,其插值、限制算子虽然没有加入光滑过程,但是使用  $K$ -循环的 Krylov 子空间方法,加速各矩阵层次上的光滑过程,同样表现出快速收敛的特性.

随着 AMG 研究的深入,其矩阵粗化方法的创新和改进层出不穷,如代数光滑误差修正的  $AMGe$ <sup>[51]</sup>, $\rho AMGe$ <sup>[52]</sup>、启发式的面中心修正格式<sup>[53]</sup>、自适应光滑聚类多重网格( $\alpha SA$ )方法<sup>[54]</sup>、自适应代数多重网格( $SA$ )方法<sup>[55]</sup>、基偏移(basis shifting approach)型光滑聚类算法<sup>[56]</sup>等.在众多的代数多重网格及其并行化方法中,仅仅针对 Poisson 方程或 Laplace 方程的数值求解,但是现实的 CFD 计算问题要复杂得多,对非结构网格下的并行开发仍然需要一段时间.文献[25]中调研了 4 种可行的并行 AMG 方法:最小区域块、 $PMIS$  粗化、光滑聚合、 $K$ -循环.在求解弱可压流的 CFD 工业应用中,从算法复杂度、计算时间上进行对比,得到  $K$ -循环方法表现出最优

的收敛特性.在文献[57]中,通过模拟 3D 汽油发动机的 CFD 问题,对比了光滑聚合型、K-循环型、经典 AMG 这 3 种方法,却得出不一致的结论:相对于其他两种方法,并行化的光滑聚合型方法并行效率较高.这也从一个侧面反映出,由于工程应用具体问题的不同,并行 AMG 的适用性也不同,寻求普适性的并行 AMG 解算器,或是针对具体问题选取更优化的并行 AMG 解算器,仍然是一个开放性的课题.

### 2.3 GMG vs. AMG

理论上,GMG 和 AMG 两者在时间复杂度和空间复杂度上都为  $O(n)$ ,因而两者性能上的差异往往体现在不同实际问题的应用中.求解问题的不同、网格层次的差异,都有可能改变两者性能上的优劣.例如,文献[58]中对比了 GMG 与 AMG 用于边数据结构有限元分析的性能,指出,在使用相同光滑算子及固定网格规模的前提下,单次迭代的 GMG 运行时间明显小于 AMG.具体而言,GMG 的启动阶段用于构建几何网格层次,AMG 启动阶段用于构建多层系统矩阵.表 1 中显示,GMG 启动时间要比 AMG 长,但是用于迭代计算的时间远远小于 AMG,因而总计算时间 GMG 相对较小.而在文献[59]中却得到不一样的结论,其求解的物理问题为自适应网格上的连通耗散系统方程,网格层次由自适应算法生成.以文献中的算例来说,将各项参数综合起来显示在表 2 中可知,在同等条件下,AMG 总迭代次数一般少于 GMG;相应地,AMG 的计算时间就比较少.但是,在表 2 的最后一个算例中,却显示出 GMG 具有较好的性能.可见,GMG 与 AMG 的性能因求解问题不同而不同.

Table 1 Comparison of calculation time<sup>[58]</sup>

表 1 计算时间对比<sup>[58]</sup>

问题规模	操作名称	CPU time (s)	
		GMG	AMG
200 800	构建网格层次	0.58	0.02
	构建限制矩阵	0.45	0.04
	构建系统矩阵	0.47	0.64
	启动时间总和	1.5	0.7
	迭代时间	3.69	13.65
	总时间	5.19	14.35
1 566 912	构建网格层次	5.02	0.17
	构建限制矩阵	3.91	0.24
	构建系统矩阵	3.81	5.85
	启动时间总和	12.74	6.26
	迭代时间	35.6	189.06
	总时间	48.34	195.32

Table 2 Comparison of iteration<sup>[59]</sup>

表 2 迭代次数对比<sup>[59]</sup>

算例	层次	规模/节点数		迭代次数	
		GMG	AMG	GMG	AMG
Problem 1 规则网格	3	1 089	1 089	27	7
	2	289	479	26	7
	1	81	331	16	6
Problem 1 自适应网格	4	1 275	1 275	22	8
	3	649	580	27	9
	2	320	304	18	8
Problem 2 规则网格	1	176	166	17	7
	3	1 089	1 089	187	256
	2	289	500	142	106
Problem 2 自适应网格	1	81	288	50	25
	4	1 046	1 046	19	142
	3	645	461	22	172
	2	381	254	25	127
	1	203	127	21	110

此外,两者的适用范围也存在一定的差异.GMG 可用于线性系统和非线性系统,需要预知和存储物理问题的网格层次结构以及各网格层次的系统离散方程,在满足网格层次特定属性的前提下,具有最优的收敛特性.相

对而言,AMG 只适用于线性系统的代数方程,对网格的几何结构以及物理背景并不关注,每次迭代过程都需要重新计算粗化过程,启动时间比较长,但是对复杂外型结构、各向异性网格、无网格等问题仍然适用,在这一点上弥补了 GMG 不能使用的缺陷.因而,对两种方法优劣的评价不能一概而论.

## 2.4 光滑算子

无论是 GMG 还是 AMG,一个比较重要的组成部分是光滑算子.光滑算子的作用在于迅速地消除各个层次上的高频误差,达到快速收敛的目的.以 GMG 来说,其光滑算子的选取相对比较灵活,高频振荡误差的消除在很大程度上与物理几何意义有关,其有效性严重依赖于光滑算子的健壮性.对于 CFD 的非结构网格问题而言,传统意义上的多重网格光滑算子,如 Jacobi, Gauss-Seidel 等,已经不能满足流场模拟在收敛速度和计算时间上的需求;另一方面,为了保证并行多重网格方法具有良好的并行可扩展性,必须减少各处理器之间的数据交换和通信开销.因此,在传统的光滑算子基础上,针对非结构网格的具体应用环境,开发出众多的光滑算子.这些光滑算子大体上可以分成 4 类:块型、线型、投影型和混合型.

### 2.4.1 块型光滑算子

这类光滑算子主要是基于全耦合系统方程,如 Navier-Stokes 方程,将原始矩阵化为块状结构,每个块对应于原始光滑算子的点进行计算,使用块内部整体推进的思想,大幅度提高了收敛速度,这样就形成了代数系统下的块 Jacobi、松弛块 Jacobi、块 Gauss-Seidel、块 SSOR 以及流体计算的块 LU-SGS<sup>[60]</sup>等光滑算子.

另外,结合上一节中的 AMG 网格粗化方法,形成了代数系统类的光滑算子,如基于 C-F 分裂光滑<sup>[29,61]</sup>、多项式类光滑<sup>[29]</sup>、稀疏近似逆<sup>[62]</sup>等多种类型光滑算子.针对其代数系统的系数矩阵,同样可以构造出块型结构的光滑算子.然而,这类方法在现代 CFD 工业应用中还不多见,其数值特性也有待下一步的应用和检验.

### 2.4.2 线型光滑算子

在对高雷诺数粘性流体的数值模拟中,将产生各向异性的系统特性,此时需要对多重网格方法的光滑算子进行适当的修正,结合半隐式粗化的线型光滑算子几乎成为标准的选择.在结构网格中,很容易实现线型的粗化光滑;而在非结构网格中,由于没有坐标方向上的标识,就需要构造类似的线型方向.文献[23]中采用权重图技术来建立各向异性网格存储结构,在强耦合方向上使用较大的权值,用于影响粗网格层次的构建,在多重网格方法中使用线隐式型光滑算子,计算粘性流体时获得与纵横比(aspect ratio)无关的收敛特性.

线隐式光滑算子<sup>[24]</sup>是基于高精度格式的  $p$ -Multigrid 方法,线型结构的构建是基于计算网格和相关物理特性的,强耦合方向对应于强连通网格的各向异性方向,通过相互之间的线性连接,构建出块三对角型的耦合线性方向的集合,在数值实验上,与经典迭代 Jacobi 相比,显示出较强的有效性.然而,由于非结构网格的特殊性,对各向异性问题的多重网格光滑算子的选取以及并行求解分析和应用,仍然值得进一步的研究.

### 2.4.3 投影型光滑算子

以 Krylov 子空间为主的投影方法<sup>[21]</sup>包括 GMRES, CG, FCG, GCR, BICGSTAB 等,其求解可以看作是在  $m$  维 Krylov 的搜索子空间中寻找近似解的过程.它实际上是一种多项式近似型方法,在时空复杂度上都具有优良的特性.表 3 中显示了部分经典迭代方法与 Krylov 子空间投影方法性能方面的差异, Krylov 子空间型的投影方法在计算时间上具有稳定数值特性,在用于 Laplace 及 Poisson 方程的求解中都显示出比经典迭代方法快的收敛特性.

Table 3 Comparison of optimal complexity for typical smoother

表 3 典型光滑算子最优复杂度比较

计算方法	类型	时间复杂度	空间复杂度	适用条件
Jacobi	基本迭代	$O(n^2)$	$O(n)$	谱半径小于 1
Gauss-Seidel	基本迭代	$O(n^2)$	$O(n)$	谱半径小于 1
SOR	基本迭代	$O(n^{3/2})$	$O(n)$	满足 $0 < \omega < 2$ , 且谱半径小于 1
CG	投影方法	$O(n^{3/2})$	$O(n)$	对称 Lanczos 型线性系统
GMRES	投影方法	$O(n^{3/2})$	$O(n)$	Arnoldi 型线性系统
BI-CGSTAB	投影方法	$O(n^{3/2})$	$O(n)$	Lanczos 双正交型线性系统



另外,以 Krylov 子空间方法作为多重网格的光滑算子,其应用过程中往往在其投影空间上结合预条件技术,可以发挥更优的性能,如预条件 GMRES(PGMRES)方法等.对于求解  $Ax=b$  的原型系统,以左预条件子  $P$  为例,原型系统的求解转化为  $P^{-1}Ax=P^{-1}b$ .为取得快速的收敛特性,预条件子  $P$  的选取应尽可能地与  $A$  矩阵近似,并且在并行可扩展性方面,预条件型的 Krylov 子空间投影方法具有较强的问题适应性.因此,在很多流体模拟的工程应用上,常常采用 Krylov 子空间的投影方法替代经典的迭代方法.

#### 2.4.4 混合型光滑算子

基于上述各种方法的混合型光滑算子,可以结合上述各自方法的优势,在非结构网格的实际应用中发挥更好的数值性能,以适用于大规模的并行计算环境.如文献[63]中使用的 GMRES+LU-SGS 的方法,对 M6,DLR-F4, DLR-F6 这 3 种机翼翼型的流场数值进行模拟,当处理器数目达到 16 时,仍能表现出稳定的数值特性.另外,文献[64]中详细地例举了各种并行化 AMG 光滑算子的性能,并且在将处理器规模增至 32 000 时,仍然能够保证较高的数值稳定性.这对于非结构网格上光滑算子大规模并行开发具有一定的借鉴价值.遗憾的是,在并行可扩展性方面仅仅描述了单个处理器上的迭代次数,并没有给出相应的数值精度以及加速比情况.

总体而言,由于实际问题的限制,光滑算子的优劣程度不是固定的,一个求解过程较优的光滑算子,在其他数值结果中的表现可能不太理想.这与问题的特殊性密切相关,所以针对具体实际应用的光滑算子的选取才变得有意义.另外,基于复杂非结构网格环境下的大规模流场数值模拟过程,光滑算子的选取不仅要保证数值计算的稳定性、收敛的快速性,而且要考虑方法的并行可扩展性.因而,数值实验结果只能作为一个借鉴,更深入的物理问题应用和理论论证仍然是一个艰巨的过程.

### 3 开源软件

CFD 计算环境下的非结构网格解算器软件,经过近 40 年的发展,形成了一大批商业软件,如 Fluent,UMS3D, FUN3D,NSU3D,CFD++,TAU 等.这些商业软件平台适应性强、可扩展性高、数值计算稳定,缺点就是价格昂贵,用户不能更改内部源代码,部分软件更是只针对特定的机构与人群开放.如,UMS3D 就只提供给美国公民,不利于普通用户的二次开发.相比之下,开源的 CFD 软件就是一种完全开放式代码,可以满足用户不同的开发需求,因此本节主要介绍几种带多重网格解算器的非结构网格开源软件.

#### 3.1 Code\_Saturne:有限体积分的通用型流体力学解算器

Code\_Saturne 是法国电力集团 EDF 公司开发的一种通用型 CFD 软件,用于求解 2D/3D 流体的 NS 方程.源代码使用 Fortran 77 编写,支持 C99 和 Python,适用于 Linux 和所有的 UNIX 平台.程序功能包括有限体积分的空间离散格式,支持结构网格、非结构网格、混合网格,能够完成定常或非定常、可压或不可压、层流或湍流、热传导或绝热等情况下的流体计算.其中,湍流计算包括混合长度、二方程模型、大涡模拟(large eddy simulation,简称 LES)等.其并行实现采用 MPI(message passing interface)标准,数值解算器中引入了 AMG,以获得快速的收敛效果.其工业应用主要集中于辐射热传导、燃烧问题、磁能动力模拟等方面.其缺点表现在程序代码结构陈旧,注释多为法文,可读性较差.更多详情可参见 <http://code-saturne/info>.

#### 3.2 OpenFVM:非等温不可压二相流的有限体积分解算器

由美国 OpenFVM 小组开发、用于求解采用有限体积分离散的湍流不可压 NS 方程,源代码使用标准 C 语言编写,可兼容 Linux 和 Windows 两大计算平台.主要功能包括空间离散采用迎风一阶分裂格式或中心差分分裂格式,时间推进包含显式与隐式两种方法,部分耦合求解,如速度-压力耦合的 SIMPLE,SIMPLER,PISO 等.网格生成采用开源的 Gmsh 软件,同时引入了网格重排序技术缩减矩阵带宽,如 RCM(Reverse Cuthill-McKee)算法,系数矩阵同样采用 CSR 格式存储,便于使用第三方数值软件 Laspack 进行求解,在并行化的数值求解时使用可扩展性能较强的 PETSc.这两个软件包中都集成了 AMG 方法.OpenFVM 的主要缺点在于空间离散格式比较简单,数值精度较低,仅适用于非等温不可压的流体计算.下载地址参见 <http://openfvm.sourceforge.net/>.

### 3.3 OpenFOAM:并行扩展性最强的流体开源软件

OpenFOAM 是目前为止应用范围最广、可扩展性最强、解算器最全的开源软件包,由美国 OpenCFD 机构开发,源代码采用标准 C++编写,可以应用于 Unix,Linux,Windows,MacOSX 等多种操作系统中.软件主要特征包括离散格式使用有限体积方法,求解可压、不可压的 Euler,NS 方程,支持多套网格和多种边界条件,如四面体、塔体、六面体等,可以实现层流定常、非定常,湍流的  $k$ -epsilon, $k$ - $\omega$ ,LES(large eddy simulation),Saplart-Allmaras 等模型的流体模拟,空间离散格式可选两种迎风格式,时间离散格式可选 Euler 向后差分 and Crank-Nicholson 差分方式,不可压 NS 方程的求解可使用 SIMPLE、PISO、耦合 Pimple 等压力-速度耦合方法,提供多种第三方软件集成,如 Metis,ParMetis,OpenMPI,ParaView,ParMGridGen 等,其中,ParMGridGen 是 GMG 的网格层次构建软件,采用结块式(agglomeration)方式并行构建网格层次,并且在数值求解中还集成了对 AMG 的并行求解的支持.

此外,自 2006 年以来,针对 OpenFOAM 的开发与应用,CFD 的开源联盟每年要组织一次国际会议进行讨论.可见,OpenFOAM 不仅在解算器上比较全面,而且对新方法的集成更新也是比较迅速的,其并行可扩展性能仅次于商业 Fluent,这也是其他开源软件不可比拟的.其负面影响是,在软件集成度越来越高、数值方法越来越全的同时,增加了软件包的负载,为多数初学者进行二次开发带来一定的难度.更多详情及下载地址可参见 <http://www.openfoam.com/>.

以上仅列举了几个典型的非结构网格解算器开源软件.其他一些比较优秀的开源软件在并行解算器上也使用了 AMG 或 GMG 方法,如 ADFC,GPDE,NUM3SIS,Featflow,Gerris,Typhon Solver 等,由于篇幅关系不一列举.总体而言,非结构网格的开源软件发展非常迅速,应用前景广阔,越来越多研究者的加入,使得软件开发呈现一片欣欣向荣的景象.同时还应看到,其开发过程也面临不少的困难,如缺乏资助、手册文档不详细、更新周期较长,应用范围较小,开发团体比较分散,涉及不同行业不同方向的研究,往往也只是实现了某类商业软件某一个方面的应用,尤其是数值解算器开发比较少,对新方法的集成不尽如人意.因此,开源 CFD 软件在给研究者提供平台的同时,也需要对其进一步加以完善.

## 4 研究展望

经过几十年的发展,非结构网格的解算器在其高效性上取得了许多重要成果.由于其致力于不断提高数值计算方法的效率,极大地推动了非结构网格工程应用的发展.在众多高效算法中,多重网格方法具有较强的数值稳定性、问题适应性以及多项式时间内收敛的高效性,成为非结构网格高效解算器必不可少的选择.两种类型的多重网格方法 AMG 和 GMG 在非结构上的应用各有侧重,很难简单地判定孰优孰劣,这需要与具体应用问题相结合.工程应用对高精度格式的需求、计算机体系结构及软硬平台的飞速发展,对非结构网格的高效解算器在并行扩展性、数值稳定性上提出了新的挑战.可以预言,以多重网格为主的非结构解算器的研究将主要集中在以下几个方面:

### (1) 结合高精度格式的 GMG 并行化应用

以离散 Galerkin、谱体积、谱差分方法为代表的高精度计算格式,结合 GMG 表现出了高效的收敛特性,在一定程度上降低了计算时间开销,如  $p$ -Multigrid 和  $hp$ -Multigrid,在求解小规模 Euler,Navier-Stokes 方程时具有较高的数值精度和较强的问题适应性能,而在大网格规模、复杂流体的并行化扩展能力方面的应用还不多见,这方面的相关文献相对较少.因而在维持解算器高阶精度以及快速收敛特性的前提下,探索并行大规模多重网格方法的有效应用,将是高精度格式下一步需要解决的问题.

### (2) 改进并行网格粗化插值过程

无论是 AMG 还是 GMG,在过去的 20 年内,网格粗化及并行化的过程一直都是一个很活跃的研究方向,涌现出一大批网格粗化插值算子.其主要原因在于,多重网格及其并行化方法的有效性,在很大程度上依赖于粗化过程中构建的网格层次结构以及插值过程引入的误差校正.粗化插值算子不仅需要数值上的稳定性,而且必须满足并行效率上的扩展性.因此,无论是代数系统矩阵的经典分裂或是光滑聚合,还是几何形体的结块聚合,有效地粗化插值算子都需要较小的计算量、快速的收敛性、较低的通信量.同时,开发普适性的粗化插值过程还

具有一定的难度,因而,结合具体实际应用的开发和优化就显得尤为重要.

### (3) 多重网格光滑算子的选取

光滑算子在 AMG 和 GMG 的有效性上发挥着重要的作用,非结构网格流体的数值模拟不仅需要光滑算子的高效性,而且必须保证其数值稳定性,在分区并行时,具有较小的通信计算比.有效的并行光滑算子不仅能够适应问题的物理特性,而且能够满足并行计算时低存储、低通信、高效率的要求.因此,设计针对具体问题并行多重网格的光滑算子,仍然是一个比较热门的话题.

### (4) 并行可扩展性应用

一般意义上的并行可扩展性应用主要关注于如何减小通信计算比.以多重网格方法为例,在循环迭代过程中常常使用计算换通信的方式获得较低的通信计算比,从而具有较高的并行效率和加速比,如多次循环一次通信、一次循环单层通信等方式,达到减少了通信量的目的.但是在减少通信的同时,数值精度却得不到保证,计算时间反而增加,并行加速比也不会很高.因此,采用启发式规则处理数值精度与通信频率的关系,也是多重网格解算器需要面对的问题.

### (5) 新型计算平台上的实现

随着多核体系结构和多核处理器技术的飞速发展,现代多核处理器已不再是将单处理器简单地拼凑在一起.越来越多的新型多核体系结构采用基于软件管理的局部存储器,将大量单处理器核整合在一起,取代了典型的基于硬件管理的 Cache 层次结构.与此同时,随着传统的图形处理部件 GPU 的计算能力和可编程性的不断提高,利用 GPU 完成通用计算任务的 GPGPU 计算成为研究的热点.这种 CPU+GPU 的异构计算模式在发掘计算机潜在的性能和性价比方面有着显著的优势,因此受到开发人员的日益青睐.应对新型体系结构,面临的一个迫切问题是哪种编程模型和编译方法对提高性能最有效.因而,探索具有较强并行可扩展性、支持复杂硬件环境的数值计算方法,仍然是今后一段时间内非结构网格解算器并行化发展的重要方向.同时,针对现有高效解算器的异构计算环境下的应用,也将被不断地开发.

## 5 结束语

总体来看,以多重网格方法为代表的非结构网格高效解算器,在其串行和并行计算过程中能够大幅度地缩短计算时间,从而显著提高求解大规模流体模拟的效率.虽然对高效解算器的研究已持续了很多年,但由于工程应用中实际问题的不断涌现,其开发始终受到广泛关注.并且随着应用问题和计算机体系结构的日新月异,将有越来越多的问题有待进一步探讨.

## References:

- [1] Mavriplis DJ. Unstructured-Mesh discretizations and solvers for computational aerodynamics. *AIAA Journal*, 2008,46(6): 1281–1298. [doi: 10.2514/1.34681]
- [2] Cockburn B, Shu CW. The Runge-Kutta discontinuous galerkin method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 1998,141(2):199–224. [doi: 10.1006/jcph.1998.5892]
- [3] Wang ZJ. Spectral (finite) volume method for conservation laws on unstructured grids: Basic formulation. *Journal of Computational Physics*, 2002,178(1):210–251. [doi: 10.1006/jcph.2002.7041]
- [4] Liu Y, Vinokur M, Wang ZJ. Spectral difference method for unstructured grids I: Basic formulation. *Journal of Computational Physics*, 2006,216(2):780–810. [doi: 10.1016/j.jcp.2006.01.024]
- [5] Solchenbach K, Trottenberg U. On the multigrid acceleration approach in computational fluid dynamics. In: *Proc. of the 4th Int'l DFVLR Seminar on Foundations of Engineering Sciences on Parallel Computing in Science and Engineering*. Bonn: Springer-Verlag, 1988. 145–158.
- [6] Yang UM. Parallel algebraic multigrid methods—High performance preconditioners. In: Bruaset A, Tveito A, eds. *Proc. of the Numerical Solution of Partial Differential Equations on Parallel Computers*. Heidelberg, Berlin: Springer-Verlag, 2006. 209–236. [doi: 10.1007/3-540-31619-1\_6]

- [7] Hülsemann F, Kowarschik M, Mohr M, Rüde U. Parallel geometric multigrid. In: Bruaset AM, Børstad P, Tveito A, eds. Proc. of the Numerical Solution of Partial Differential Equations on Parallel Computers. Heidelberg, Berlin: Springer-Verlag, 2005. 165–208.
- [8] Mavriplis DJ, Darmofal D, Keyes D, Turner M. Petaflops opportunities for the NASA fundamental aeronautics program. In: Proc. of the 18th AIAA Computational Fluid Dynamics Conf. Miami: AIAA, Inc., 2007. 25–61.
- [9] Komatitsch D, Erlebacher G, Göddede D, Michéa D. High-Order finite-element seismic wave propagation modeling with MPI on a large GPU cluster. *Journal of Computational Physics*, 2010,229(20):7692–7714. [doi: 10.1016/j.jcp.2010.06.024]
- [10] Tölke J, Krafczyk M. TeraFLOP computing on a desktop PC with GPUs for 3D CFD. *Int'l Journal of Computational Fluid Dynamics*, 2008,22(7):443–456. [doi: 10.1080/10618560802238275]
- [11] Diskin B, Thomas JL. Comparison of node-centered and cell-centered unstructured finite-volume discretizations: Inviscid fluxes. In: Proc. of the 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. Orlando: AIAA, 2010. 1–24.
- [12] Diskin B, Thomas JL, Nielsen EJ, Nishikawa H, White JA. Comparison of node-centered and cell-centered unstructured finite-volume discretizations, Part I: Viscous fluxes. *AIAA Journal*, 2009,48(6):1326–1338. [doi: 10.2514/1.44940]
- [13] Jameson A, Schmidt W, Turkel E. Numerical solution of the euler equations by finite volume methods using Runge-Kutta time-stepping schemes. In: Proc. of the 14th Fluid and Plasma Dynamics Conf. Palo Alto: AIAA, 1981. 41–60.
- [14] Jameson A. Solution of the Euler equations by a multigrid method. *Applied Mathematics and Computation*, 1983,13(2):327–356.
- [15] Shu CW. Total-Variation-Diminishing time discretizations. *SIAM Journal on Scientific and Statistical Computing*, 1988,9(6): 1073–1084. [doi: 10.1137/0909073]
- [16] Gottlieb S. On high order strong stability preserving Runge-Kutta and multi step time discretizations. *Journal of Scientific Computing*, 2004,25(1):105–128. [doi: 10.1007/s10915-004-4635-5]
- [17] Wang HH. A parallel method for tridiagonal equations. *ACM Trans. on Mathematical Software*, 1981,7(2):170–183. [doi: 10.1145/355945.355947]
- [18] Rasetarinera P, Hussaini MY. An efficient implicit discontinuous spectral galerkin method. *Journal of Computational Physics*, 2001,172(2):718–738. [doi: 10.1006/jcph.2001.6853]
- [19] Saad Y, Schultz MH. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 1986,7(3):856–869. [doi: 10.1137/0907058]
- [20] Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 1997,131(2):267–279. [doi: 10.1006/jcph.1996.5572]
- [21] Saad Y. *Iterative Methods for Sparse Linear Systems*. 2nd ed., Philadelphia: SIAM Press, 2003. 205–230.
- [22] Duff IS, Uçar B. Combinatorial problems in solving linear systems. In: Naumann U, Schenk O, Simon HDm Toledo S, eds. *Dagstuhl Seminar Proc. Dagstuhl: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik*, 2009. 1–39.
- [23] Mavriplis DJ. Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes. *Journal of Computational Physics*, 1998,145(1):141–165. [doi: 10.1006/jcph.1998.6036]
- [24] Fidkowski KJ, Oliver TA, Lu J, Darmofal DL.  $p$ -Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 2005,207(1):92–113. [doi: 10.1016/j.jcp.2005.01.005]
- [25] Emans M. Performance of parallel AMG-preconditioners in CFD-codes for weakly compressible flows. *Parallel Computing*, 2010, 36(5):326–338. [doi: 10.1016/j.parco.2009.12.009]
- [26] Parthasarathy V, Kallinderis U. A new multigrid approach for 3D unstructured, adaptive grids. *AIAA Journal*, 1994,32(5):956–963. [doi: 10.2514/3.12080]
- [27] Mavriplis DJ. A 3D agglomeration multigrid solver for the reynolds-averaged navier-stokes solver for unstructured meshes. *Int'l Journal for Numerical Methods in Fluids*, 1996,23(1):527–544.
- [28] Bonhaus DL. An upwind multigrid method for solving viscous flows on unstructured triangular meshes. [MS. Thesis]. Ohio: The School of Engineering and Applied Science, University of Cincinnati, 1993.
- [29] Burgerscentrum JM. Iterative solution methods. *Applied Numerical Mathematics*, 2011,51(4):437–450. [doi: 10.1016/j.apnum.2004.06.003]

- [30] Müller JD, Giles MB. Edge-Based multigrid schemes for hybrid grids. In: Proc. of the 4th Numerical Methods for Fluid Dynamics. ICFD, 1998. 425–432.
- [31] Lallemand MH, Steve H, Dervieux A. Unstructured multigriding by volume agglomeration: Current status. *Computers & Fluids*, 1992,21(3):397–433. [doi: 10.1016/0045-7930(92)90047-y]
- [32] Kernighan BW, Lin S. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 1970,49(1): 291–307.
- [33] Pothen A, Simon HD, Liu KPP. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 1990,11(3):430–452. [doi: 10.1137/0611030]
- [34] Karypis G, Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 1998,20(1):359–392. [doi: 10.1137/s1064827595287997]
- [35] Karypis G, Kumar V. Parallel multilevel  $k$ -way partitioning scheme for irregular graphs. In: Proc. of the 1996 ACM/IEEE Conf. on Supercomputing. Pittsburgh: IEEE Computer Society, 1996. 35–56. [doi: 10.1145/369028.369103]
- [36] Mavriplis DJ, Aftosmis MJ, Berger M. High resolution aerospace applications using the NASA columbia supercomputer. In: Proc. of the 2005 ACM/IEEE Conf. on Supercomputing. IEEE Computer Society, 2005. [doi: 10.1109/sc.2005.32]
- [37] Moulitsas I, Karypis G. Multilevel algorithms for generating coarse grids for multigrid methods. In: Proc. of the 2001 ACM/IEEE Conf. on Supercomputing. Denver: ACM Press, 2001. 45–55. [doi: 10.1145/582034.582079]
- [38] Teresco JD, Devine KD, Flaherty JE. Partitioning and dynamic load balancing for the numerical solution of partial differential equations. In: Bruaset A, Tveito A, eds. Proc. of the Numerical Solution of Partial Differential Equations on Parallel Computers. Heidelberg, Berlin: Springer-Verlag, 2006. 55–88. [doi: 10.1007/3-540-31619-1\_2]
- [39] Chrisochoides N. A survey of parallel mesh generation methods. In: Proc. of the Computational Science and Engineering. Heidelberg, Berlin: Springer-Verlag, 2006.
- [40] Nastase CR, Mavriplis DJ. High-Order discontinuous galerkin methods using a spectral multigrid approach. In: Proc. of the 43rd AIAA Aerospace Sciences Meeting and Exhibit. Reno: AIAA, 2005. 12–24. [doi: 10.1016/j.jcp.2005.08.022]
- [41] Luo H, Baum JD, Löhner R. A  $p$ -multigrid discontinuous Galerkin method for the Euler equations on unstructured grids. *Journal of Computational Physics*, 2006,211(2):767–783. [doi: 10.1016/j.jcp.2005.06.019]
- [42] Cleary AJ, Falgout RD, Henson VE, Jones JE. Coarse-Grid selection for parallel algebraic multigrid. In: Proc. of the Workshop on Parallel Algorithms for Irregularly Structured Problems. Heidelberg, Berlin: Springer-Verlag, 1998. 104–115. [doi: 10.1007/BFb0018531]
- [43] Vaněk P, Mandel J, Brezina M. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 1996,56(3):179–196. [doi: 10.1007/bf02238511]
- [44] Henson VE, Yang UM. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 2002,41(2):155–177. [doi: 10.1016/S0168-9274(01)00115-5]
- [45] Sterck HD, Yang UM, Heys JJ. Reducing complexity in parallel algebraic multigrid preconditioners. *SIAM Journal on Matrix Analysis and Applications*, 2004,27(4):1019–1039. [doi: 10.1137/040615729]
- [46] Krechel A, Stüben K. Parallel algebraic multigrid based on subdomain blocking. *Parallel Computing*, 2001,27(8):1009–1031. [doi: 10.1016/s0167-8191(01)00080-1]
- [47] Falgout RD, Yang UM. hypre: A library of high performance preconditioners. In: Sloot P, Hoekstra A, Tan C, Dongarra J, eds. Proc. of the Int'l Conf. on Computational Science (ICCS 2002). Amsterdam: Springer-Verlag, 2002. 632–641. [doi: 10.1007/3-540-47789-6\_66]
- [48] Tuminaro RS, Tong C. Parallel smoothed aggregation multigrid :aggregation strategies on massively parallel machines. In: Proc. of the Supercomputing, ACM/IEEE 2000 Conf. Washington: IEEE Computer Society, 2000. 5–25. [doi: 10.1109/SC.2000.10008]
- [49] Notay Y, Vassilevski PS. Recursive Krylov-based multigrid cycles. *Numerical Linear Algebra with Applications*, 2008,15(5): 473–487. [doi: 10.1002/nla.542]
- [50] Gee MW, Siefert CM, Hu JJ, Tuminaro RS. ML 5.0 smoothed aggregation user's guide. Technical Report, SAND2006-2649, Albuquerque: Computational Mathematics and Algorithms, Sandia National Laboratories, 2007. 3–66.

- [51] Brezina M, Cleary AJ, Falgout RD, Henson VE, Jones JE. Algebraic multigrid based on element interpolation (AMGe). *SIAM Journal on Scientific Computing*, 2000,22(5):1570–1592. [doi: 10.1137/s1064827598344303]
- [52] Chartier T, Falgout R, Henson VE, Jones JE, Vassilevski PS, Manteuffel TA, McCormick SF, Ruge JW. Spectral element agglomerate AMGe. In: Widlund OB, Keyes DE, eds. *Proc. of the Domain Decomposition Methods in Science and Engineering XVI*. Heidelberg, Berlin: Springer-Verlag, 2007. 513–521. [doi: 10.1007/978-3-540-34469-8\_64]
- [53] Jones JE, Vassilevski PS. AMGE based on element agglomeration. *SIAM Journal on Scientific Computing*, 2001,23(1):109–133. [doi: 10.1137/s1064827599361047]
- [54] Brezina M, Falgout R, MacLachlan S, Manteuffel T, McCormick S, Ruge J. Adaptive smoothed aggregation ( $\alpha$ SA) multigrid. *SIAM Review*, 2005,47(1):317–346. [doi: 10.1137/050626272]
- [55] Brezina M, Falgout R, MacLachlan S, Manteuffel T, McCormick S, Ruge J. Adaptive algebraic multigrid. *SIAM Journal on Scientific Computing*, 2006,27(4):1261–1286. [doi: 10.1137/040614402]
- [56] Gee MW, Hu JJ, Tuminaro RS. A new smoothed aggregation multigrid method for anisotropic problems. *Numerical Linear Algebra with Applications*, 2009,16(1):19–37. [doi: 10.1002/nla.593]
- [57] Emans M. AMG for linear systems in engine flow simulations. In: Wyrzykowski R, Dongarra J, Karczewski J, Wasniewski K, eds. *Proc. of the Parallel Processing and Applied Mathematics*. Heidelberg, Berlin: Springer-Verlag, 2010. 350–359. [doi: 10.1007/978-3-642-14403-5\_37]
- [58] Watanabe K, Igarashi H, Honma T. Comparison of geometric and algebraic multigrid methods in edge-based finite-element analysis. *IEEE Trans. on Magnetics*, 2005,41(5):1672–1675. [doi: 10.1109/TMAG.2005.846092]
- [59] Wu CT, Elman HC. Analysis and comparison of geometric and algebraic multigrid for convection-diffusion equations. *SIAM Journal on Scientific and Statistical Computing*, 2007,28(41):2208–2228. [doi: 10.1137/060662940]
- [60] Parsani M, Abeele KVd, Lacor C, Turkel E. Implicit LU-SGS algorithm for high-order methods on unstructured grid with  $p$ -multigrid strategy for solving the steady Navier-Stokes equations. *Journal of Computational Physics*, 2009,229(3):828–850. [doi: 10.1016/j.jcp.2009.10.014]
- [61] Falgout RD, Vassilevski PS. On generalizing the AMG Framework. *SIAM Journal on Matrix Analysis and Applications*, 2004,42(4):1669–1693. [doi: 10.1137/s0036142903429742]
- [62] Tang WP, Wan WL. Sparse approximate inverse smoother for multigrid. *SIAM Journal on Matrix Analysis and Applications*, 2000,21(4):1236–1252. [doi: 10.1137/s0895479899339342]
- [63] Luo H, Sharov D, Baum JD. Parallel unstructured grid GMRES LU-SGS method for turbulent flows. *AIAA Journal*, 2003,23(2):22–37.
- [64] Baker AH, Falgout RD, Kolev T, Yang UM. Multigrid smoothers for ultra-parallel computing. *SIAM Journal on Scientific Computing*, 2010,33(5):2864–2887. [doi: 10.1137/100798806]



李宗哲(1983—),男,湖南常德人,博士生,主要研究领域为 CFD 的数值并行计算方法.  
E-mail: lzz144@163.com



姚路(1983—),男,博士生,主要研究领域为并行计算,图形分割,矩阵排序.  
E-mail: shaoeq@163.com



王正华(1962—),男,博士,教授,博士生导师,主要研究领域为高性能计算应用,系统性能评价.  
E-mail: zhhwang188@sina.com



曹维(1983—),男,博士生,CCF 学生会员,主要研究领域为 GPGPU 计算,并行 Boltzmann 方法.  
E-mail: caoweil93@gmail.com