DOI:10.16262/j.cnki.1000-8217.1996.01.002

·学科发展·

多重网格算法综述

李晓梅　莫则尧

(国防科技大学计算机系，长沙 410073)

［摘要］ 本文总结了多重网格算法的基本物理背景、各部分应用准则以及已取得的丰硕应用成果，仔细探讨了影响经典多重网格算法并行效率的几个瓶颈问题，以及最新理论框架下多重网格算法所赋予的新的生命力和广阔应用前景。基于当前并行计算特点，展望了多重网格并行计算的几个有意义的尝试方向。

［关键词］ 多重网格算法，迭代，频率分解，套迭代，粗网格校正，并行计算

1 引言

在大量工程与物理问题中，偏微分方程是最重要的数学模型。对这一类方程的求解，往往采用有限差分、有限元或有限体积等离散方法将求解区域划分成网格，然后求解所得网格方程。由于受到舍入误差、计算机内存和计算复杂度的限制，直接求解网格方程几乎是不可能的，实际中一般采用迭代法。早期的迭代方法，如 Guass-Seidel，Jacobi，ILU，ADI，与直接法一样，存在一个缺陷：工作量与未知量个数不成正比。而人们往往坚信这样一条金科玉律[3]：计算量应该与物理量个数成正比，任何出现延迟的数值方法都不是最优的。

多重网格算法正是满足以上规律的一种迭代方法。它兴起于最近 20 年，最初可追溯到 Southwell[5]的两层网格松驰方法，然后 Fedorenko 和 Bachvalov[5]将其推广到多层，并指出潜在的快速收敛性，但当时并没有引起人们的足够注意。进入 70 年代中期，A. Brandt[2]和 W. Hackbusch[18]的工作才标志着多重网格算法研究的全面开始。在其后的十多年内，E. Bank，Fuchs，Hemker，Jameson，McCormick，Stüben，Trottenberg，Wessling 等学者做了大量工作，将多重网格算法应用到各个领域，揭示基本原理，力争最优数值效率。相关的国际出版物、论文与日俱增。80 年代的成果，比较系统地总结在 A. Brand[3]，W. Hackbusch[18]，W. Briggs[7]，S. McCormick[27]和 P. Wessling[31]的专著中，那里包含了上千篇文献。至此，多重网格算法基本成熟，只是收敛性证明远远落后于实际计算的效果，尤其对于比较复杂的问题。

经过近 20 年的发展，多重网格算法已经成为数值计算领域中一种加速迭代收敛的技术，一门新的学科，而不仅仅是一种单纯的算法。尤其进入 90 年代后，由于 O. Widlund[32]，

本工作获国家自然科学基金和 863-306 高科技计划资助.

本文于 1995 年 9 月 15 日收到.

J. Bramble[1]，J. Xu[33,34]等人的努力，视所有迭代方法为子空间校正，将多重网格算法融入新的理论框架中，使得以前棘手的收敛性证明在这里变得相对容易，并与区域分解算法融为一体，二者仅子区域的划分不同，从而使得传统多重网格技术焕发出强大生命力和应用前景，尤其在并行计算机上的应用。多重网格算法，无论串行和并行，都是当今数值计算领域最活跃的分支之一。

由于多重网格算法的数值高效性，人们从来没有停止过对它的并行化研究。从标量机到向量机，从 SIMD 型机器到 MIMD 型机器，所有的并行计算研究机构都将其作为一个重要课题。发现了经典多重网格算法并行化过程中几个难以克服的瓶颈问题，并采取了相应措施。当今并行计算朝大粒度 MPP 方向发展，如何提高多重网格算法的并行效率，仍是一个艰巨任务。可喜的是，与区域分解技术的融合，为并行多重网格算法开辟了新的天地。

本文注重于多重网格算法的应用效果，从浩如烟海的文献中总结了多重网格算法的基本物理背景、各部分应用准则以及已取得的应用成果，仔细探讨了多重网格算法并行计算的瓶颈问题。针对当前流行的并行计算特点，提出了将来并行化手段的几种可行方案。

2　基本原理与应用规则

多重网格算法的最初动力来源于对网格方程迭代求解时，误差的各个 Fourier 分量的不同衰减程度。认识到高频振荡误差是局部行为，来源于附近几个网格点之间的相互耦合，与边界或距离较远的网格点信息无关；而低频光滑误差是全局行为，主要来源于边界信息。传统的点或块松弛都是局部性较强的方法，因此它们能迅速抹平局部性的高频振荡误差，但对全局性的低频光滑误差却衰减缓慢。实际上，经过初始几次迭代后，误差将呈现光滑性。所以，习惯上称能迅速抹平高频振荡误差，使误差趋于光滑的松驰方法为有效光滑方法，并用松驰因子[2]来刻画它们的光滑效应。

既然通过局部松驰后误差呈现光滑性，则此时误差主要来源于边界。可以设想二维 $N\times N$ 网格上的点松驰方法，将边界信息传播到所有点至少需 $O(N)$ 次迭代，因此收敛速度奇慢。不妨将网格方程的剩余部分（残差）限制到粗网格上进行。在粗网格上精确求解后，将所得解延拓到细网格上，与原来近似解组合，形成网格方程的近似解，称这一过程为粗网格校正。在粗网格上，由于网格点少，边界信息能较快地传播到所有网格点，收敛速度将加快。同样地，在粗网格上也存在高、低频误差，类似于细网格，进行几次局部松弛消除高频误差后，可以将低频误差再转移到更高层网格。如此进行下去，直到最高层网格，那里未知量个数非常少，直接精确求解的工作量可忽略不计。然后从高层到低层依次将所得解返回、组合，在最细网格上最终形成一个近似解，这一递归性质称为套迭代技术。多重网格算法就是这样将问题的求解分布在不同的层上，所有层相互协调地求解同一问题的。实际上，此时的粗网格校正就是起到将边界信息迅速传播到所有网格点的作用。这一结构非常类似于树或金字塔形网络，那里的高层节点主要负责在叶子节点间快速传递信息。

细网格松弛、粗网格校正和套迭代技术是多重网格算法的三大支柱。细网格松驰负责消除高频振荡误差，粗网格校正负责消除低频光滑误差，套迭代技术负责通过限制和延拓算子连接所有层共同求解同一问题。可见多重网格算法的基本思想是：（1）一个问题能在不同规模的网格上求解；（2）细网格仅仅只需负责消除高频误差，而粗网格负责消除低频误差。

当然，松弛方法、限制和延拓算子可能相互耦合高、低频误差。对应某个低频，存在一个高频调和空间[31]，包含所有将会耦合的频率，它们在算法运行过程中相互影响。我们应该仔细地选择多重网格算法的每个部分，尽量避免这种耦合带来的危害。

多重网格算法除了具有数值高效性外，还有一个优点，即可以分解成几个相对独立的部分，程序设计模块性好，结构清晰，可重用率高。下面，我们就从算法对物理问题离散格式、松弛方法、限制与延拓、网格粗化策略以及套迭代技术的需求，分别阐述应用多重网格算法的基本准则。

离散格式 离散格式必须是稳定的。格式的数值稳定性是一种局部行为，将导致解的局部高频振荡，使得松弛方法在细网格上无法有效光滑误差，从而使多重网格算法的效率将很低，甚至发散。而离散解光滑部分的稳定性一般仅仅依赖于微分方程本身的稳定性，与离散格式关系不大。当然，物理问题本身的稳定性是一个基本前提。

由于离散格式的稳定性是局部行为，因此可以用局部模方法[3]来分析。要求离散格式能将方程本身在一个或几个网格步长上的所有振荡较好地描述出来，以便于松弛方法有效地消除。但是，对给定的步长 h，只判断格式是否稳定是不够的，还应该给出一个尺度来刻画这种稳定性的程度。h-椭圆性[3]就是一个很好的判别方法。具有 h-椭圆性的离散格式，都必定存在一种有效的松弛方法，消除所有高频误差。

对于椭圆型方程，符合 h-椭圆性的离散格式是存在的。而对于奇异扰动问题、非椭圆型问题，可引入人工粘性来获得满足 h-椭圆性的离散格式，或者根据信息传播的方向定义强耦合方向以及 s-h-椭圆性等等[3]，只是松弛时采用相应措施。

松弛方法 总原则是让残差从细网格限制到粗网格之前，充分光滑。光滑效应的度量采用光滑因子 μ，可通过局部模分析方法简单地获取。若在每层网格上松弛 υ 次，则可用 μ^{υ} 来预测多重网格算法的近似最优收敛因子。松弛时要注意以下几点：

(1) 每层网格上松弛次数不宜过多（一般 2—3 次），只需有效消除高频误差，因为松弛到一定程度，低频误差将占主导地位，与高频误差相互耦合，无助于限制之前残差的光滑。

(2) 最好具有数值稳定性，避免迭代过程中的高频振荡。

(3) 点松弛方法仅光滑最强耦合方向的误差，对存在次耦合方向的问题，如各向异性问题、奇异扰动问题，则必须采取块迭代方法，如线松弛、平面松弛，使位于同一块的所有未知量被同时松弛或者使强耦合方向的未知量被同时松弛。

(4) 光滑因子不宜过小，一般使得多重网格算法的收敛因子在 0.1 左右为最佳。红黑序的 Guass-Seidel 迭代是一个较好的选择。

(5) 光滑方法尽量追求 Robust 性，使多重网格算法适应多种类型的问题。

(6) 对通常的偏微分方程系统，只需对其主部和次主部考虑光滑效应。

限制与延拓 主要考虑算子的阶，它们依赖于原始方程中导数的阶。设含有 q 个未知函数的 q 个微分方程，m_{ij} 表示第 j 个未知函数在第 i 个方程中的最高导数的阶。设这 j 个未知函数在限制和延拓过程中相互独立。令 m^j 表示第 j 个未知函数的延拓阶，m_i 表示第 i 个方程残差限制阶。则应有以下关系式成立

(1) 低频调和空间中高频通过粗网格校正振幅被放大

$$1+O\left(\sum_{i,j} h^{m_i+m^j-m_{ij}}\right)$$

倍。故为了避免由于粗网格校正引起高频振荡，应有 $m_i+m^j \geq m_{ij}$，同时也可以看出，$m_i+m^j > m_{ij}+1$ 是不必要的。

(2) 细网格限制时，每个高频对低频振幅的贡献为 $O(h^{m_i-m_{ij}})$，故应有 $m_i \geq m_{ij}$ 对那些高、低频相互耦合的松弛方法，如红—黑序 Gauss-Seidel 迭代，还应有

$$m_i > \sum_k (m_{ij}-r_{kj})$$

其中 $O(r_{kj})$ 为高频误差在第 j 个方程由于松弛所产生的对低频的贡献。

(3) 对完全多重网格算法 (FMG)，第 j 个未知函数从粗网格到细网格第一次延拓的阶 $\overline{m}_j$ 必须满足 $\overline{m}_j \geq p+l_j$，以保证细网格上第一次出现的高频误差的阶不小于微分方程的离散阶 p，其中 l_j 为第 j 个方程微分算子的阶。

另外，还有边界近似阶的影响等等，请参看文献 [3，31]。对对称正定椭圆型算子，一般使得限制与延拓算子相互共轭。

网格粗化策略　从程序设计的模块化、易移植性出发（尤其对并行机），一般采用标准网格粗化策略（步长在所有方向扩大一倍）。但为了保持点松弛，对某些特殊问题，可以采用半粗化策略，即仅粗化次耦合方向网格，使之也变为强耦合方向。粗网格上方程离散格式和迭代方法都可以与细网格上的不同，对对称正定算子，典型代表 为 Galerski 近似。

套迭代技术　一般采用 V 或 W 循环，W 循环能保持收敛因子不随网格层的变化而变化，具有 Robust 性，但代价相对昂贵；当网格层不多时，V 循环具有同样的性质，但计算量小，因此更受欢迎。除此之外，还存在 S 循环[21]、F 循环[23]，它们的性能介于 V 循环与 W 循环之间，视具体情形可分别采用。

可以看出，针对不同的问题，多重网格算法的各个部分可以变化，相互组合，以达到最优。只有把握了基本原理，从模型问题到实际复杂问题进行系统的数值实验，仔细选择各个部分，才能对不同的问题获得令人满意的与步长 h 无关的收敛效果。现在，理论上的证明远远跟不上实际应用的效果，有待人们的进一步探索。这里我们只对多重网格算法的一些基本性质进行了总结，至于具体到非线性问题[3,18,31]，如何追求 Robust 性，与自适应技术的结合[3,22]，快速组合多重网格技术 (FAC)[26]，非椭圆型问题[5]等等，请参看相关文献。

3　已取得的成果和待扩充领域

本部分总结主要源于文献 [5，7，27，31]，并只讨论经典串行应用。

多重网格算法经过近 20 年的研究，在经典应用领域——线性和非线性、标量和非标量椭圆型问题取得了丰硕的成果，每个未知量只需十多次算术运算便能在误差允许范围内正确求解，具有内在并行性，并且能适应这些情况：自由边界问题，狭小区域问题，各种类型的奇异与非连续性，局部网格细化，严重非线性问题，区域中含有粗网格上不可见的小洞问题，高振荡边界问题等等。在弹性力学，网格生成技术方面也取得了同样的效率。

类似于椭圆型问题，从 80 年代开始，多重网格算法已深入到计算流体力学 (CFD)，时间相关问题、波动方程、积分方程等领域[3,18,31]。对流体力学 Euler 方程、Stokes 方程、Navier-Stokes 方程、位势方程中的大量定常与非定常问题，加入人工粘性，采用适当的离散格式

（如迎风格式，矢通量分裂格式，半离散格式）后，能以同样的效率求解。在牺牲一至几个数量级效率条件下，可以成功求解含单族或多族特征线的高雷诺数不可压缩流问题[5]。

多重网格算法在求解来自波动方程的高不适定问题时，效果非常不理想，但如果转化为积分问题，则效果要好得多。同样，这种转换也适合求解特征值或特征函数问题。

对时间相关抛物型问题，不但在求解隐式离散格式时与椭圆型算子具有同样的效率，而且只要问题本身在时间方向上光滑，则求解过程在时间方向访问细网格的次数很少，能迅速达到稳定状态。

多重网格算法被推广到别的领域，取得了大量成果，如统计物理中的快速 Monte-Carlo 方法，积分变换，人工智能中 N 个体的相互关系识别，全局优化问题，图象处理，量子色动力学（QCD）[5]等等。同时，多重网格技术与别的领域中高效方法结合，产生了许多新方法，如高精度谱多重网格算法[36]，处理非规则问题的代数多重网格方法[4]，与有限元结合的协调、非协调元多重网格算法[35]，非结构网格上多重网格算法[9]等等。

4　经典多重网格并行计算的几个瓶颈问题

多重网格算法发展的 20 年，也是并行计算机飞速发展的 20 年。从标量机到向量机，从共享存储多处理机系统到分布式存储多处理机系统，一直到当今流行的 MPP 系统和工作站机群，都吸引着人们将许多有效的串行程序移植到各种类型的机器上。由于多重网格算法的数值高效性，自它诞生以来，人们就致力于其原理并行度探讨[6,30,10]，在向量机[19]、SIMD 类型机器[25]、超立方体[8]等各种类型并行计算机系统上从模型问题到复杂问题、从简单区域到复杂区域、从椭圆型问题到时间相关问题进行了大量的研究。但早期的工作比较凌乱，基于原理性和细粒度 PRAM 并行计算模型的讨论比较多，离实用还存在一段距离。

系统地探讨多重网格并行化工作得归功于德国的 GMD[24]。他们在各种类型的机器，包括 Cray Y-MP，Intel ipsc/860，Meiko，SUPRENUM，Ncube，CM-2，CM-5 等工作站机群上，对经典多重网格算法的并行化进行了大量的实验，提出了一些有效实现方法，仔细探讨了多重网格算法的并行效率。这些工作主要包含在文献［16，25，28，29］中。

目前，多重网格算法成熟的并行化工作主要集中于串行程序的并行化，在尽量不影响算法收敛性前提下考虑并行，这样能保持原有算法的数值高效性。并行化方法主要可分为两种：对规则区域采用网格划分，将原始网格划分成几块，分配给各台处理机，并且相互包含相邻网格块的拟边界信息；对复杂区域采用块结构方法，根据几何外形将区域划分成几块到几十块，每块适当变换成规则区域，在这些规则子区域上进行多重网格的计算。大量试验和理论结果表明，经典多重网格算法的并行化工作存在以下几个瓶颈问题。

（1）经典多重网格算法本身属于 Gaùss-Seidel 类型算法，层与层之间的计算相互关联。

（2）对分布式存储系统，当处理机台数较多时，粗网格上通讯所占的比重将远远大于计算量，并行效率损失严重。

（3）细网格松弛过程中，边界信息更换次序的混乱可能导致精度的污染。如果保持串行边界信息的更换次序，则频繁的等待与信息交换也将影响效率的发挥。

（4）算法可伸缩性差，属于中度并行问题，只适于在几台或十多台处理机上运行。

尤其在进入当前每台处理机功能很强的并行计算时代，问题（1）和问题（2）将显得更

为严重。虽然人们对以上瓶颈问题进行了大量改进，例如 P. Frederickson 的并行超收敛技术[14]，文献 [17] 中的减少粗网格访问次数，但都不能改变经典多重网格算法的中度并行性质。那么，对于大规模并行计算，多重网格算法的并行效率将是一个非常令人担忧的问题。

5 多重网格算法的大规模并行计算

当前并行计算朝协同方向发展[15]，即大量同构或异构型功能较强的处理单元通过高性能网络相互连接，协同地求解大规模科学与工程计算问题。其典型代表为 MPP 和工作站机群。一般具有以下特点：(1) 分布式存储。(2) 拥有大量处理单元，几十到几百个甚至上千个不等，每个处理单元功能较强，类似于当前流行的工作站核心部分，每秒几千万次到几亿次甚至几十亿次浮点结果。(3) 拥有高性能互连网络。

在这样一个环境下，实践证明高效率的获取一般通过以下途径：(1) 数据并行或区域分解：将任务按区域进行分割，分配给各台处理机完成。计算过程中处理机间可进行适当通讯，局部通讯比全局通讯更受欢迎。但只要全局通讯不频繁，也不会成为并行效率的障碍。(2) 大粒度并行，相对增加数值计算比重。而影响并行效率的关键因素为：(1) 负载平衡；(2) 通讯与负载的比例；(3) 计算与通讯的重叠，屏蔽通讯延迟时间。

针对以上并行计算特点，获取较高的经典多重网格算法并行效率难度比较大，因此必须寻求新的途径，与当前流行的另一数值方法：区域分解算法有效结合。

区域分解算法[20]是随着并行计算机的产生而产生的数值计算方法。它将问题的求解区域划分成几个或几十个相互重叠或不重叠子区域，分配给各台处理机。这些子区域相互包含相邻区域的拟边界信息，相互迭代共同求解同一问题。早期典型代表为 Schwarz 类型算法，具有很好的局部性，负载平衡能力强，并行效率高，程序设计简单。但是收敛速度与各子区域间重叠区域大小相关，数值效率低，当子区域个数较多时，这种现象更为明显。O. Widlund[32] 指出，类似于这种没有任何全局信息交换的区域分解算法，迭代条件数至少为 $O(\frac{1}{H^2})$，其中 H 为所有子区域直径的最大值，即随着子区域个数的增加，条件数呈平方增长，这无疑给大规模并行计算带来困扰，迫切要求出现条件数与子区域个数无关的区域分解算法。为此，早期有 J. Bramble[1]等人的迭代子结构方法，实际上为非重叠区域分解算法，程序设计稍微复杂。后来出现了 Dryja 与 O. Widlund 针对对称正定问题提出的叠加型 Schwarz 算法[12]，或小区域重叠型区域分解算法[13]，其条件数与子区域个数无关，且适合于大规模并行。

多重网格与区域分解的结合可分为两种策略：第一种，利用多重网格算法近似求解子区域问题。这是很自然的想法，并行效率依赖于区域分解算法，而数值效率来源于二者的结合。关键技术在于如何进行全局通讯以及保持负载与通讯的平衡。

第二种策略，改进现有经典 Guass-Seidel 型多重网格算法，使之变为 Jacobi 型多重网格算法。这一工作得归功于 J. Bramble，J. Xu 等学者的子空间校正算法。对对称正定问题，他们利用子空间校正思想将所有迭代方法统一在一个理论框架内，并给出了理论上的统一证明。所有迭代方法被归纳为两类：连乘型 (Guass-Seidel 型) 和叠加型 (Jacobi 型)。求解区域被划分为许多不同的子空间，所有子空间上的问题都可以单独求解，并通过共轭梯度法相互联系求解原始问题。连乘型算法的各个子空间问题求解时相互关联，收敛性较好，适合于串行

机；而叠加型算法的各个子空间问题可以独立求解，收敛性稍差，但并行度高，适合于大规模并行机。于是，多重网格算法在理论上得到升华，与别的迭代算法，如区域分解，本质上是一致的，仅仅为子空间划分不同，许多人习惯上称此时的多重网格算法为多水平(multilevel)方法。目前，这种新的理论框架正在被推广到非对称问题、不适定问题和其它领域。自然，多重网格算法也会得到新的发展。

视多重网格算法的每层网格为求解区域的一个有限元离散，则问题可在不同层网格上求解。若采用连乘型算法，则类似于经典多重网格技术，层与层之间相互关联；若采用叠加型算法，则所有层网格上的问题可独立并行求解，都可以视为原来问题解在该层上的校正，最后由共轭梯度法综合所有校正值，获得原始问题的近似解。已经证明，不管哪种算法，收敛因子都与未知量个数和网格层数无关。这样，我们可以根据并行计算机的特点，选择不同的子空间，甚至将每层网格再划分成多个子区域，采用叠加型算法，可以实现大规模并行。这方面的典型代表为X. Zhang的多水平Schwarz算法[37]，它是区域分解与多重网格的一种结合方式。从这一理论框架出发，充分考虑到并行计算中负载、通讯平衡，并行与数值效率的权衡，实现的难度，计算与通讯的重叠等各个方面，能够突破经典多重网格算法的中度并行特性，在大规模并行计算机上获得很好的综合效率。

参考文献

[1] Bramble J, Pasciak J, Xu j. Parallel Multilevel preconditioners. Math. Comput., 1990, **55** (191): 1—22.

[2] Brandt A. Multilevel adaptive solutions to boundary value problems. Math. Comput., 1977, **31**.

[3] Brandt A. Multigrid Thchniques: 1984 Guide, with applications to fluid Dynamics, GMD-Studien. Nr. 85, 1984.

[4] Brandt A. Algebric Multigrid Theory: The Symmetric Case. Appl. Math. and Comput., 1986, **19** (1—4); 23—56.

[5] Brandt A. Multiscale computational methods: research activities, Multigrid Course '92, PB93-133916, 1992.

[6] Brandt A. Theoretical parallel multigrid, Multigrid Course '92, PB93-133916, 1992.

[7] Briggs W. Multigrid tutorial, SIAM, Philadelghia, 1987.

[8] Chan T, Saad Y. Multigrid algorithms on the hypercube multigrid processors, IEEE Trans. Comput., 1986, **c-35** (11), 969-977.

[9] Chan T. Multigrid and Domain Decomposition Methods for Elliptic Problems on Unstructured Meshes, The 8th intern. conf. on DDM, 1995, Beijing, China.

[10] Douglas C C. A Review of Numerous Parallel Multigrid Methods, SIAM News, 1992, **25** (3).

[11] Douglas C C. MG Net Bibliography, October, 1995.

[12] Dryja M, Widlund O. Towards a unified theory of domain decomposition algorithms for elliptic problems. In Third International Symposium on Domain Decomposition Methods for Partial Differeftial Equations, T. Chan, R. Glowinski, J. Pèriaux and O. Widlund eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[13] Dryja M, Widlund O. Domain decomposition algorithms with small overlap. SIAM J. Sci. Comput., 1994, **15** (3): 604-620.

[14] Frederickson P, MaBryan O. Parallel superconverged multigrid, in Proceedings of the Third Copper Mountain Conference on Multigrid Methods, S. McCormick ed., Marcel Dekker, NY, 1987, pp. 195-210.

[15] Fox G, Williams R, Messina P. Parallel Computing Works. Morgan Kaufmann Publischers, Inc., San Francisco, California, 1994.

[16] Gartel U, Ressel L. Parallel multigrid: Grid Partitioning versus Domain Decomposition, Arbeitspapiere der GMD, Nr. 599, 1991.

[17] Gupta S N, Zubair M, Grosch C. A multigrid algorithm for parallel computers: CPMG, J. Sci. Comput., 1992, **7** (3).

[18] Hackbush W. Multigrid methods and applications, Springer Verlag, 1985.

[19] Holter W. A vectorized multigrid solver for the three dimensional poisson equation. Appl. Math. and Comput., 1986, **19**, (1—4): 127—144.

[20] 吕涛. 区域分解算法，北京：科学出版社，1992.

[21] Lyengar S, Goyal A, Comparison of S and V cycles in multigrid methods for linear elliptic equations with variable coefficients. Numerical Methods for Partial Differential Equations, 1992, **8**; 113-125.

[22] van der Maarel H T M. Adaptive multigrid for the steady Euler equations. Comm. in Appl. Numer. Methods, 1992, **8**: 749—760.

[23] Mandel J. On the Multigrid F-Cycle. Applied Math. and Comput., 1990, **37**: 19-36.

[24] McByan O. The SUPRENUM and GENESIS projects. Parallel Computing, 1994, **20**: 1389—1396.

[25] McByan O, Frederickson P, Linden J et al. Multigrid methods on parallel computers——A survey of recent developments. Impact of computing in science and engineering, 1991, **13**: 1—75.

[26] McCormick S, Thomas J. The Fast Adaptive Composite Grid (FAC) methods for elliptic equations. Math. Comput., 1986, **46** (174): 439—456.

[27] McCormick S ed., Multigrid methods, Frontiers in applied mathematics, Vol. 5, SIAM, Philadelphia, 1987.

[28] Ritzdorf H, Schüller A et al. L_iSS——An environment for the parallel multigrid solution of partial differential equations on general 2D domains. Parallel Computing, 1994, **20**: 1559—1570.

[29] Schieweck F. A parallel multigrid algorithm for solving the Navier-Stokes equations. impact of computing in science and engineering, 1993, **5**: 345—378.

[30] Tuminaroa R, Womble D. Analysis of the multigrid FMV cycle on large-scale parallel machine. SIAM J. Sci. Comput., 1993, **14** (5); 1159—1173.

[31] Wesseling P. An introduction to multigrid methods. Pure and Applied Mathematics series, John Wiley and Sons, 1992.

[32] Widlund O. Iterative substructuring methods: Algorithms and theory for elliptic problems in the plane, in First Intern. Symp. on DDM for partial differential equations, R. Glowinski, G. H. Golub, G. A. Meurant *et al.* eds., SIAM, Philadelphia, 1988.

[33] Xu J. Theory of multigrid methods, Ph. D. thesis, Cornell University, Ithaca, NY, Rep AM-48, Pennsylvannia State University, University Park, PA, 1989.

[34] Xu J. Iternative methods by space decomposition and subspace correction, SIAM Reviem, 1992, **34** (4): 581-613.

[35] 蔚喜军. 协调和非协调有限元多重网格算法，博士论文，中科院计算中心，1994.

[36] Zang T, Hussaini M. On the spectural multigrids for time-dependent Navier-Stokes Equations, Appl. Math. and Comput., 1986, **19** (1—4).

[37] Zhang X. Multilevel Schwarz Methods. Numer. Math., 1992, **63**: 521—539.

VIEWPOINTS OF MULTIGRID ALGORITHMS

Li Xiaomei Mo Zeyao

(Dept. of Computer, National University of Defence, Changsha 410073)

Abstract The basic physical backgrounds, applied rules and rich applied results of multigrid algorithms are surveyed in this paper. At the same time, we investigate the bottleneck problems to decrease the parallel efficiency of multigrid algorithms, discuss the latest or future developments of parallel mulgrid computations under the new uniform theory.

Key words multigrid algorithms, iterations, coarser grid correction, parallel computing