ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, Supplement, December 2009, pp.8−14

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

HYPRE中多重网格解法器的并行可扩展性能分析*

徐小文+, 莫则尧, 曹小林

(北京应用物理与计算数学研究所 高性能计算中心,北京 100094)

Parallel Scalability Analysis for Multigrid Solvers in HYPRE

XU Xiao-Wen+, MO Ze-Yao, CAO Xiao-Lin

(High Performance Computing Center, Institute of Applied Physics and Computational Mathematics, Beijing 100094, China)

+ Corresponding author: E-mail: xwxu@iapcm.ac.cn

Xu XW, Mo ZY, Cao XL. Parallel scalability analysis for multigrid solvers in HYPRE. *Journal of Software*, 2009,20(Suppl.):8−14. http://www.jos.org.cn/1000-9825/09002.htm

Abstract: HYPRE is a high performance preconditioners library for solving large sparse linear systems on parallel computers. This paper analyzes the scalability of SMG and BoomerAMG, both are multigrid solvers in HYPRE, on massively parallel computer with thousands of processors. Based on the presented results, some conclusions are got to design scalable algorithms and their parallel implementation in real numerical applications.
Key words: HYPRE; linear solver; preconditioner; iterative method; multigrid; parallel computing; scalability

摘 要: 测试并分析了高性能预条件库 HYPRE 的多重网格解法器 SMG 和 BoomerAMG 在某国产大规模并行机数千个处理器上的可扩展性能,得到若干对线性解法器算法研究和并行实现技术发展具有启示性意义的结论.这些结论对实际复杂物理系统数值模拟中线性解法器的应用和发展具有一定的指导意义.
关键词: HYPRE;线性解法器;预条件子;迭代方法;多重网格;并行计算;可扩展性能分析

在很多实际数值模拟应用中,线性解法器作为共性层面的基本模块对提高数值模拟性能具有重要作用.HYPRE(high performance preconditioners)[1]是美国Lawrance Livemore国家实验室(LLNL)开发的面向大规模并行机的高性能预条件库,包含了数十种迭代方法和预条件子的并行实现,在实际复杂物理系统的数值模拟中得到广泛应用.目前,HYPRE已经发布了数十个版本,本文的描述基于 2006 年 12 月发布的 2.0.0 版本.

HYPRE 最重要的特征之一是为不同的应用提供不同的接口,用户可按照自己表示线性代数方程组的方式向 HYPRE 提供数据.目前,HYPRE 提供 4 个接口:

- Struct 接口:面向结构网格离散的应用.每个网格点的离散格式具有相同的模式,目前该接口仅支持标量偏微分方程,每个网格点仅允许定义单个物理量.
- Sstruct 接口:面向半结构(semi-struct)网格离散的应用.所谓半结构网格是指具有某些非结构特征的结构网格,如结构网格自适应计算中的 AMR 网格和多块结构网格(multi-blocks).该接口可支持偏微分方程组,每个网格点允许定义多个物理量.

* Supported by the National Natural Science Foundation of China under Grant Nos.90718029, 60603050, 10701015 (国家自然科学基金); the National Basic Research Program of China under Grant No.2005CB321702 (国家重点基础研究发展计划(973))
Received 2008-07-01; Accepted 2009-04-02

● FEI 接口:面向有限元方法离散的应用.该接口允许用户按有限元的数据结构(如单元刚度矩阵)提供矩阵.

● IJ 接口:线性代数接口.该接口以矩阵方式显式地表示线性代数方程组,是适用范围最广泛的接口,在上述 3 个接口都不适用时,该接口是唯一的,也是可行的选择.

此外,HYPRE 为不同的接口定义了不同的数据结构,并配以适合该接口的解法器.我们在表 1 中列出了目前 HYPRE 提供的部分解法器及其所适用的接口.基于表 1,用户可根据自己的应用特征和需求,选择适当的接口和解法器,然后提供所选接口和解法器所需的数据结构,调用 HYPRE 进行求解.

表1 HYPRE提供的解法器及适用接口

解法器名称	解法器描述	适用接口
Jacobi	对角尺度预条件子(包括块Jacobi).	所有
SMG	半粗化多重网格解法器.二维和三维情形的光滑子(smoother)分别采用线松弛和面松弛,对三维情形的面松弛,则调用二维情形的SMG进行求解.	Struct, SStruct
PFMG	半粗化多重网格解法器.与SMG唯一不同之处在于,PFMG采用简单的点松弛作为光滑子.	Struct, SStruct
FAC	快速自适应组合网格解法器.专用于SStruct接口的基于SAMR网格的应用.	SStruct
BoomerAMG	代数多重网格解法器.通用解法器,用户可选择不同的并行粗化策略及松驰格式光滑子.	所有
SplitSolve	专门针对SStruct接口开发的解法器.该解法器将矩阵分裂为两部分:结构和非结构部分,结构部分用SMG或PFMG求解,非结构部分用BoomerAMG进行求解.	SStruct
AMS	Maxwell方程专用解法器.采用辅助子空间方法作为预条件子.	FEI, IJ
MLI	代数多重网格解法器.基于光滑聚集(smoothed aggregation)的粗化和插值策略,面向有限元的应用.	FEI
ParaSails	并行稀疏近似逆预条件子.	所有
Euclid	ILU预条件子的并行可扩展库.	所有
Krylov	Krylov子空间迭代方法类:包括PCG,GMRES,BiCGSTAB等方法.通用解法器,通常作为预条件子的加速器.	所有
Hybrid	时间发展方程专用解法器.对于时间发展方程,通常每个时间步都形成一个新的线性代数方程组,这些方程组的性质可能会随着时间的推进而发生变化.由此,Hybrid允许相应的解法器也随之调整.如,当系数矩阵从强对角占优变成弱对角占优时,相应的解法器自适应地由Jacobi-CG变成SMG-CG或BoomerAMG-CG.	所有

多重网格解法器是 HYPRE 的特色之一.HYPRE 提供多个多重网格解法器,如 AMS,SMG,PFMG,MLI, BoomerAMG.这些解法器从专用向通用过渡,期望满足各种应用的需求(见表 1).在这些解法器中,SMG 和 BoomerAMG 是目前实际应用中使用最广泛的两个解法器.

本文针对三维模型问题,测试 SMG 和 BoomerAMG 解法器在某国产大规模并行机数千个处理器上的可扩展性能力,为 HYPRE 在实际复杂物理系统数值模拟中的应用前景提供参考.

本文第 1 节介绍多重网格算法以及 SMG 和 BoomerAMG 解法器.第 2 节介绍性能测试方法和测试环境.第 3 节给出性能测试结果和分析.第 4 节基于本文的分析,得到若干结论.

1 多重网格解法器

本节先给出多重网格算法的简单描述,然后简单介绍多重网格解法器 SMG 和 BoomerAMG.

1.1 多重网格算法

考虑如下偏微分方程离散得到的线性代数方程组:

$$A^h u^h = f^h \tag{1}$$

其中, $A^h \in R^{n\times n}$ 是系数矩阵; $u^h, f^h \in R^n$,分别是解向量和右端.

多重网格(multigrid,简称MG)[2]是求解方程组(1)最有效的迭代方法之一.其主要思想是在迭代过程中利用简单松弛迭代方法的光滑特性消除误差的高频分量,利用粗网格校正消除剩下的低频分量,两者相互配合从而达到整体误差快速下降的目的.在很多情形下,多重网格方法可被证明具有最优或近似最优的计算复杂度,即收敛速度与问题规模无关,计算量与问题规模成线性或近似线性关系.多重网格方法的迭代过程可由算法 1 描述.

算法 1. 多重网格迭代 $MG(A^h,u^h,f^h,\mu_1,\mu_2)$.

每个多重网格迭代执行如下步骤:

(1) 在细网格 Ω^h 上进行前光滑:对 $A^hu^h=f^h$ 做 μ_1 次松弛迭代.

$$u^h \leftarrow S^{\mu_1}Su^h+\sum_{j=0}^{\mu_1-1}S^jB^{-1}f^h.$$

(2) 粗网格校正:

(2.1) 残差向量限制: $f^H \leftarrow I_h^H(f^h-A^hu^h)$;

(2.2) 在粗网格 Ω^H 上求解粗网格方程 $A^Hu^H=f^H$:

如果 Ω^H 是最粗网格层,则精确求解 $A^Hu^H=f^H$.

否则递归执行 $MG(A^H,u^H,f^H,\mu_1,\mu_2)$;

(2.3) 校正近似解: $u^h \leftarrow u^h+I_H^hu^H$.

(3) 在细网格 Ω^h 上后光滑:对 $A^hu^h=f^h$ 做 μ_2 次松弛迭代.

$$u^h \leftarrow S^{\mu_2}u^h+\sum_{j=0}^{\mu_2-1}S^jB^{-1}f^h.$$

在算法 1 中, μ_1, μ_2 分别为前后光滑中松弛迭代次数, $S=I-B^{-1}A$ 为相应的迭代矩阵,称为光滑算子, B 为预条件子,一般采用 Jacobi 或 Gauss-Seidel 松弛迭代,即 $B=D_A$ 或 $(D-L)_A$,其中, D_A 和 $(D-L)_A$ 分别为矩阵 A 的对角和下三角部分. $I_h^H:\Omega^h\to\Omega^H$ 和 $I_H^h:\Omega^H\to\Omega^h$ 分别为细网格到粗网格的限制算子和粗网格到细网格的插值算子.

根据粗网格 Ω^H 的构造方法,多重网格通常可分为几何多重网格(GMG)和代数多重网格(AMG)[3]两种方法.GMG方法中,粗网格是事先存在的,或可以基于细网格的几何信息进行构造.AMG方法则是基于代数信息,如矩阵元素符号和大小等来构造粗网格.关于多重网格方法的详细讨论,可参见文献[4].

1.2 SMG和BoomerAMG解法器

SMG 和 BoomerAMG 是 HYPRE 的两个主要多重网格解法器.下面给出这两个解法器的简单介绍:

● SMG:半粗化多重网格(semi-coarsening MG)解法器[5].所谓半粗化是指SMG构造粗网格时,仅沿其中一个方向进行粗化(如 x 方向),而在其他方向上(如二维情形的 y 方向或三维情形的 yz 平面)进行线松弛(二维)或面松弛(三维).因此,SMG是一个基于结构网格应用的特殊几何多重网格解法器.特别地,线松弛中HYPRE采用循环归约方法求解三对角方程,面松弛中则调用二维情形的SMG方法进行求解.需要注意的是,虽然SMG事先存在网格层次结构,但其粗网格算子是通过Galerkin方法计算的(与AMG一样),因此,类似地存在一个启动阶段(setup)来构造限制和插值算子以及粗网格算子.只是SMG通过构造特殊的插值算子来保证粗网格算子与细网格算子具有相同的格式宽度和矩阵结构,从而保持网格的几何信息.

● BoomerAMG:并行代数多重网格解法器[6].AMG区别于几何多重网格的主要特征之一在于粗网格层并不是事先存在的.因此,在启动阶段,AMG需要一个独立的粗化过程(coarsening)通过代数途径构造粗网格层.不同的粗化策略对AMG算法性能和并行实现性能有较大影响,具体地,粗化策略的优劣可通过如下指标衡量:

- 收敛速度:通过 AMG 迭代达到收敛阈值的迭代次数 *nits* 来衡量.

- 算子复杂度 C_A:各层网格算子 A^l 的非零元数目之和与最细层网格算子 A^1 的非零元数目的比值,即 $C_A=\left(\sum_l|A_l|\right)\Big/|A_1|$. C_A 可在一定程度上反映单次迭代的计算量和存储开销.

BoomerAMG实现了当前文献中近 10 种粗化策略,不同的粗化方法具有不同的侧重点.大体而言,这些粗化策略可分成两种类型[7]:

- P_1 型粗化:具有较快的收敛速度,但具有较高的算子复杂度.如 CLJP,Falgout 等粗化属于该类型.

- P_1' 型粗化:具有低算子复杂度性质,但算法收敛速度有所下降.如 PMIS,HMIS 等粗化属于该类型.

2 测试方法和环境

2.1 测试方法

我们将解法器的可扩展性能分为如下两个方面[8]:

- 算法可扩展能力:刻画计算复杂度依赖于问题规模和(或)计算资源的程度.计算复杂度可通过单次迭代计算量和达到收敛的迭代次数来反映,与机器和具体实现无关.
- 并行可扩展能力:刻画单次迭代的并行效率.取决于算法并行度、并行实现技术和并行机性能.

我们采用弱可扩展模型,即固定单进程的问题规模而增加进程数目,并结合文献[8]中的分析方法对上述两种能力进行分析和评价.具体地,对于每一个评价阶段,设 T_1 和 T_p 分别表示处理器数目为 1 和 p 时的 CPU 时间,则在弱可扩展模型下,该阶段的计算效率 E_p 为

$$E_p = \frac{T_1}{T_p} \tag{2}$$

进一步地,由文献[8]可知,计算效率可分解为算法效率和并行效率,即

$$E_p = E_{alg} \times E_{par} \tag{3}$$

对于某阶段的计算,如果能够给出算法效率的可度量表达式,则可以得到上述分解的数值刻画.对于 SMG 和 AMG,其算法都可分为启动和求解两个阶段.对于这两个独立的阶段,利用现有的指标均可对算法效率进行评价,具体地:

- 对 SMG 解法器,启动阶段的算法效率 $E_{set-alg}$ 可以认为近似等于 1.0,则计算效率等于并行效率.求解阶段的算法效率 $E_{sol-alg}$ 则可由迭代次数进行衡量,即

$$E_{sol-alg} = \frac{nits_1}{nits_p} \tag{4}$$

其中, $nits_1$ 和 $nits_p$ 分别为处理器数目为 1 和 p 时的迭代次数.

- 对BoomerAMG解法器,启动和求解阶段的算法效率 $E_{set-alg}$ 和 $E_{sol-alg}$ 分别如下刻画:

$$E_{set-alg} = \frac{C_A^1}{C_A^p},\ E_{sol-alg} = \frac{C_A^1}{C_A^p} \times \frac{nits_1}{nits_p} \tag{5}$$

其中, C_A^1 和 C_A^p 分别为处理器数目为 1 和 p 时的算子复杂度.

2.2 测试环境

某国产大规模并行计算机,包含数千个处理器.测试中,程序采用二级编译优化选项.

3 测试结果和分析

本节先给出测试模型,然后分别给出解法器 SMG 和 BoomerAMG 的性能测试结果和分析.

3.1 测试模型

本文以如下三维标量 Poisson 方程为测试模型:

$$-\Delta u(x) = f(x) \tag{6}$$

其中, $x \in \Omega = [0,1]^3$,Dirichlet 边界条件,采用 7 点格式进行离散.我们记该问题为 3D-7pt 模型问题.

3.2 SMG性能分析

首先考查 SMG 的存储开销可扩展能力.表 2 给出了处理器数 p =1 和 p =4 096 两种情形下 SMG 求解 3D-7pt 模型问题规模上限的近似值.由此可以看出:

- 当 p =4 096 时,SMG 求解规模可达 20 亿个网格点,单进程平均规模可超过 50 万个网格点.
- 与 p =1 的情形相比, p =4 096 时的平均单进程计算能力低了约 1 个量级.

表2 SMG求解3D-7pt模型问题的计算规模上限

p=1	p=4 096	
	总规模	单进程平均规模
6.8×10^6	2.0×10^9	5.12×10^5

表 3～表 4 给出了固定单进程规模分别为 40^3,120^3的可扩展性能结果.其中,T_{sol},T_{set},T_{tot}分别为求解阶段、启动阶段和总过程的计算效率;$E_{sol\text{-}alg}$和$E_{sol\text{-}par}$分别为求解阶段的算法效率和并行效率,由公式(2)~公式(4)计算;$nits$是迭代次数;$E_{set\text{-}par}$是启动阶段的并行效率.

表3 SMG求解3D-7pt模型问题性能结果,每个进程规模为40×40×40

P	$T_{sol}(E_{sol},nits,E_{sol\text{-}alg},E_{sol\text{-}par})$	$T_{set}(E_{set}=E_{set\text{-}par})$	$T_{tot}(E_{tot})$
1	1.6(1.00,6,1.00,1.00)	0.3(1.00)	1.9(1.00)
8	3.0(0.53,7,0.86,0.62)	0.9(0.33)	3.9(0.49)
64	7.4(0.22,8,0.75,0.29)	2.8(0.11)	10.2(0.19)
512	15.2(0.11,8,0.75,0.15)	6.5(0.05)	21.7(0.09)
1 000	19.7(0.08,8,0.75,0.11)	8.0(0.04)	27.7(0.07)
4 096	34.7(0.05,9,0.67,0.07)	24.0(0.01)	58.7(0.03)

表4 SMG求解3D-7pt模型问题性能结果,每个进程规模为120×120×120

P	$T_{sol}(E_{sol},nits,E_{sol\text{-}alg},E_{sol\text{-}par})$	$T_{set}(E_{set}=E_{set\text{-}par})$	$T_{tot}(E_{tot})$
1	74.5(1.00,7,1.00,1.00)	7.6(1.00)	82.1(1.00)
8	106.9(0.70,8,0.88,0.80)	10.9(0.70)	117.8(0.70)
64	122.4(0.61,8,0.88,0.69)	14.0(0.54)	136.4(0.60)
512	180.0(0.41,9,0.78,0.53)	22.5(0.34)	202.5(0.41)
1 000	180.0(0.41,9,0.78,0.53)	28.4(0.27)	208.4(0.39)

从表 3～表 4 我们得到如下结论:

- SMG 具有良好的算法可扩展能力,随着问题规模和处理器数目的增加,迭代次数基本保持常数,算法效率基本上始终保持在 75%以上.
- 无论是求解阶段还是启动阶段,SMG 的并行可扩展能力都不理想,难以在数百个处理器上获得良好的并行效率,尤其是启动阶段的并行效率很不理想.这是导致总体可扩展能力不理想的主要原因.
- 当单进程规模特别大(超过 100 万个网格点,见表 4)时,在数百个处理器上并行效率可达到 30%～50%.
- SMG 很难在上千个处理器上获得良好的并行可扩展能力.

3.3 BoomerAMG性能分析

本节以 Falgout 和 PMIS 两种典型的粗化方法为代表来考查 BoomerAMG 的可扩展能力.前者以保证 AMG 快速收敛为准则,具有较高的算子复杂度,属于 P_1 型粗化.后者以保证低算子复杂度为准则,属于 P_1' 型粗化,具有较慢的收敛速度.因此,通常将基于 PMIS 粗化的 AMG 方法作为预条件子,通过 Krylov 子空间迭代方法加速收敛.本文中,对 PMIS 粗化采用 GMRES(10)方法进行加速,记为 PMIS-GMRES(10).

首先考查 BoomerAMG 的存储开销可扩展能力.表 5 给出了处理器数 p =1 和 p =4 096 两种情形下,BoomerAMG 求解 3D-7pt 模型问题规模上限的近似值.由此可以看出:

- 对于 PMIS 粗化,性能结果几乎与 SMG 一样(表 2),在处理器数目 p =4 096 时的求解规模可达 20 亿个网格点,单进程平均规模可超过 50 万个网格点.
- 对于 Falgout 粗化,在处理器数目 p =4 096 的平均单进程计算能力比单进程情形低了近 3 个量级.由此可以断定 Falgout 粗化无法扩展至数千个处理器.

表5 BoomerAMG求解3D-7pt模型问题的计算规模上限:Falgout vs. PMIS

粗化策略	p=1	p=4 096	
		总规模	单进程平均规模
Falgout	5.8×10^6	1.3×10^7	3.3×10^3
PMIS	8.0×10^6	2.0×10^9	5.1×10^5

导致 Falgout 粗化无法扩展的主要原因在于,基于该粗化的 AMG 算法具有不可扩展的算子复杂度,限制了其存储可扩展能力.对应于表 5 中处理器数目 p =4 096 时的计算规模,表 6 分别给出了 Falgout 粗化和 PMIS 粗化的算子复杂度的可扩展性能.从表 6 我们可以看到 PMIS 粗化的算子复杂度可扩展能力非常理想,基本保持为常数.然而,随着问题规模和处理器数目的增加,Falgout 粗化的算子复杂度呈现明显递增趋势,当处理器数目为 4 096 时,其最大的格式宽度(矩阵一行的非零元数目)竟然达到 11 263.

由于Falgout粗化无法扩展到数千个处理器,因此,下面我们仅给出PMIS的测试结果.表 7 和表 8 给出了固定单进程规模分别为 40^3,80^3的可扩展性能结果.其中T_{sol},T_{set},T_{tot}分别为求解阶段、启动阶段和总过程的计算时间;$E_{sol\text{-}alg}$和$E_{sol\text{-}par}$分别为求解阶段的算法效率和并行效率,由公式(2)~公式(4)计算;$nits$是迭代次数;$E_{set\text{-}par}$是启动阶段的并行效率.由于 PMIS 的算子复杂度具有近似最优的可扩展能力(见表 6),即 $C_A^1 \approx C_A^p$,因此,公式(5)中启动和求解阶段的算法效率分别变为

$$E_{set-a\lg} \approx 1.0,\ E_{sol-a\lg} \approx \frac{nits_1}{nits_p} \tag{7}$$

表6 对应于表5中p=4 096时的计算规模上限,BoomerAMG算子复杂度可扩展能力:Falgout vs. PMIS.其中,$E_{CA} = C_A^1 / C_A^p$ 为算子复杂度效率,$S_{mx}(lev)$为所有网格层中网格算子格式宽度的最大值S_{mx}及其出现的层号lev

P	Falgout (单进程规模15×15×15)		PMIS (单进程规模80×80×80)	
	$C_A(E_{CA})$	$S_{mx}(lev)$	$C_A(E_{CA})$	$S_{mx}(lev)$
1	3.3(1.00)	114(3)	2.3(1.00)	114(3)
8	10.7(0.31)	593(5)	2.3(1.00)	120(4)
64	19.2(0.17)	1985(7)	2.4(0.96)	131(3)
512	27.5(0.12)	4754(9)	2.4(0.96)	136(4)
1 000	30.2(0.11)	6 789(10)	2.4(0.96)	138(4)
4 096	34.0(0.10)	11 263(11)	2.4(0.96)	139(3)

表7 BoomerAMG性能结果(PMIS-GMRES(10)),每个进程规模为40×40×40

P	$T_{sol}(E_{sol},nits,E_{sol\text{-}alg},E_{sol\text{-}par})$	$T_{set}(E_{set}=E_{set\text{-}par})$	$T_{tot}(E_{tot})$
1	0.6(1.00,13,1.00,1.00)	0.3(1.00)	0.9(1.00)
8	1.0(0.60,17,0.76,0.79)	0.6(0.50)	1.6(0.56)
64	1.9(0.32,24,0.54,0.59)	0.8(0.38)	2.7(0.33)
512	3.3(0.18,34,0.38,047)	0.9(0.33)	4.2(0.21)
1 000	4.3(0.14,38,0.34,0.41)	1.3(0.23)	5.6(0.17)
4 096	5.9(0.10,47,0.28,0.36)	2.1(0.14)	8.0(0.11)

表8 BoomerAMG性能结果(PMIS-GMRES(10)),每个进程规模为80×80×80

P	$T_{sol}(E_{sol},nits,E_{sol\text{-}alg},E_{sol\text{-}par})$	$T_{set}(E_{set}=E_{set\text{-}par})$	$T_{tot}(E_{tot})$
1	8.6(1.00,17,1.00,1.00)	3.0(1.00)	11.6(1.00)
8	15.6(0.55,23,0.74,0.74)	6.7(0.45)	22.3(0.52)
64	23.1(0.37,33,0.52,0.71)	9.9(0.30)	33.0(0.35)
512	49.2(0.17,46,0.37,0.46)	11.2(0.28)	60.4(0.19)
1 000	63.6(0.14,64,0.27,0.52)	10.9(0.28)	74.5(0.16)
4 096	127.2(0.07,109,0.16,0.44)	14.2(0.21)	141.4(0.08)

从表 7~表 8 我们得到如下结论:

- 在数百个处理器上,PMIS的求解阶段可获得较理想的并行效率,当 p =4 096 时,单进程规模为 40^3和 80^3的求解阶段并行效率分别为 36%和 44%.启动阶段的并行效率不太理想.
- 算法可扩展能力下降成为影响求解阶段可扩展能力的主要因素.这说明,低算子复杂度的粗化策略导致 AMG 算法偏离了多重网格算法的最优性质.

4 结 论

以三维 Poisson 方程为模型问题,本文测试了 HYPRE 的两个多重网格解法器 SMG 和 BoomerAMG 在某国产大规模并行机上的可扩展性能.测试结果表明,SMG 解法器具有很好的算法可扩展能力,随着问题规模和处

理器数目的增加,其迭代收敛速度几乎保持常数.然而,无论是启动阶段还是求解阶段,都难以在数百个处理器上获得理想的并行可扩展能力,更无法扩展到数千个处理器.BoomerAMG 解法器中,高算子复杂度的粗化策略,如 Falgout 算法,无法扩展到上千个处理器.低算子复杂度的粗化策略,如 PMIS 算法,在上千处理器上具备良好的并行可扩展能力,然而,算法可扩展能力不太理想.

可以预料的是,在实际复杂物理系统数值模拟应用中,这两个解法器具有各自的优点和缺点:SMG 以及基于高算子复杂度的 AMG 算法能够获得快速收敛但并行可扩展能力不理想,基于低算子复杂度的 AMG 解法器则相反.因此,一方面,对实际数值模拟应用程序,需要根据实际情况,权衡解法器在算法和并行两个方面的可扩展性能;另一方面,对于解法器中迭代方法的构造,应该同时考虑数值算法的最优性以及并行实现的可扩展性,研究算法和并行可扩展均好的高效迭代方法.

References:

[1] HYPRE: High-Performance preconditioning library. http://acts.nersc.gov/hypre/main.html

[2] Brandt A. Multi-Level adaptive solution to boundary-value problems. Mathematics of Computation, 1977,31(138):333−390.

[3] Brandt A, McCormick S, Ruge J. Algebraic multigrid (AMG) for sparse matrix equations. In: Evans DJ, ed. Sparsity and Its Application. Cambridge: Cambridge University Press, 1984. 257−284.

[4] Trottenberg U, Oosterlee CW, Schuller A. Multigrid. Academic Press, 2001.

[5] Brown PN, Falgout RD, Jones JE. Semicoarsening multigrid on distributed memory machines. SIAM Journal on Scientific Computing, 2000,21(5):1823−1834.

[6] Henson VE, Yang UM. Boomer AMG: A parallel algebraic multigrid solver and preconditioner. Applied Numerical Mathematics, 2002,41:155−177.

[7] Xu XW, Research on scalable parallel algebraic multigrid algorithms [Ph.D. Thesis]. Beijing: Chinese Academy of Engineering Physics, 2007 (in Chinese with English abstract).

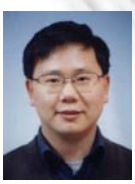
[8] Xu XW, Mo ZY, Scalability analysis for parallel algebraic multigrid algorithms. Chinese Journal of Computational Physics, 2007, 24(4):387−394 (in Chinese with English abstract).

附中文参考文献:

[7] 徐小文.可扩展并行代数多重网格算法研究[博士学位论文].北京:中国工程物理研究院,2007.

[8] 徐小文,莫则尧.并行代数多重网格算法可扩展性能分析.计算物理,2007,24(4):387−394.

徐小文(1978－),男,湖南郴州人,博士,助理研究员,主要研究领域为并行计算,多重网格方法.



莫则尧(1971－),男,博士,研究员,博士生导师,主要研究领域为并行计算,大规模数值模拟应用,软件研制.

曹小林(1974－),男,博士,研究员,主要研究领域为并行计算.