

求解大规模稀疏线性代数方程组序列的自适应 AMG 预条件策略

徐小文^{①②*}, 莫则尧^{①②}, 安恒斌^{①②}

① 北京应用物理与计算数学研究所计算物理重点实验室, 北京 100094

② 中物院高性能数值模拟软件中心, 北京 100086

* 通信作者. E-mail: xwxu@iapcm.ac.cn

收稿日期: 2016-03-20; 接受日期: 2016-07-07; 网络出版日期: 2016-10-25

国家重点基础研究发展计划 (973) (批准号: 2011CB309702) 和国家自然科学基金 (批准号: 61370067, 91430218, 91530324) 资助项目

摘要 时间相关偏微分方程隐式离散后, 通常要求解一个稀疏线性代数方程组序列. 利用序列中相邻方程组性质的差异性与相似性, 自适应地选取预条件子, 提升方程组序列的并行求解效率, 从而缩短总体求解时间, 是一个值得研究的问题. 本文针对科学与工程计算中广泛使用的代数多重网格 (AMG) 预条件子, 设计了方程组序列相关的自适应预条件策略. 通过惯性约束聚变 (ICF) 的辐射流体力学数值模拟典型应用, 验证了该策略的有效性. 测试结果表明, 在某高性能计算机的 3125 个 CPU 核上, 自适应预条件策略可将并行效率从 47% 提升到 61%, 将模拟总时间从 19.7 h 降为 14.5 h.

关键词 稀疏线性解法器 迭代方法 预条件子 代数多重网格算法 (AMG) 并行计算

1 引言

在基于时间相关的偏微分方程 (PDE) 数值模拟中, 如果时间方向采用隐格式离散, 随着时间步进以及非线性迭代向前推进, 要求解一个稀疏线性代数方程组序列:

$$\{A_i x_i = b_i, i = 1, \dots, n_{\text{eq}}\}, \quad (1)$$

其中 A_i , x_i , b_i 分别为第 i 个方程组的稀疏矩阵、未知向量和右端向量, n_{eq} 为该序列中方程组数目. 在很多数值模拟中, 典型应用的一次完整模拟所需求解的方程组数目高达数万、数十万, 甚至数百万, 求解该方程组序列的 CPU 时间占总时间的 80% 以上, 是主要性能瓶颈.

受计算和存储复杂度的限制, 迭代方法是当前大规模计算中求解该类方程组的主流方法. 迭代方法的收敛速度往往受限于问题规模和应用特征, 需要采用预条件技术 (preconditioning) 加速迭代收敛. 对于由偏微分方程离散得到的稀疏线性代数方程组, 代数多重网格 (AMG) 算法^[1~3] 是目前实际大规模计算中最有效的预条件技术之一. 传统的 AMG 算法包括两个步骤: 启动阶段 (setup) 和求解阶段.

引用格式: 徐小文, 莫则尧, 安恒斌. 求解大规模稀疏线性代数方程组序列的自适应 AMG 预条件策略. 中国科学: 信息科学, 2016, 46: 1411–1420, doi: 10.1360/N112016-00074

启动阶段基于稀疏矩阵信息, 构造多重网格算法组件, 包括: 粗网格层及粗网格矩阵、网格层间转移算子等. 求解阶段基于这些算法组件, 在网格层之间反复迭代, 执行多重网格循环 (cycle). 在每个网格层, 通过光滑子 (smoother) 将迭代误差的高频分量 (震荡部分) 消除, 将剩下的低频分量 (光滑部分) 由粗网格校正消除. 通过光滑过程与粗网格校正过程的互补, 基于 AMG 预条件的迭代方法在很多情形下具有最优或近似最优的计算复杂度, 迭代次数与问题规模无关.

尽管如此, AMG 算法的并行可扩展性是一个公开的困难问题^[4~7], 主要瓶颈在于启动阶段的粗网格矩阵构造. 在现有 AMG 算法中, 粗网格矩阵主要采用 Galerkin 方法通过 3 个稀疏矩阵的乘积来实现 ($A_c = RAP$, 其中 A 为细网格层矩阵, R 和 P 分别为细网格层与粗网格层之间的限制和插值算子, A_c 是粗网格矩阵). 相对于细网格矩阵, 虽然粗网格矩阵的规模变小, 但由 RAP 计算的粗网格矩阵无法保持细网格矩阵的稀疏结构, 随着网格不断粗化, 粗网格矩阵越来越稠密, 通信模式变得越来越复杂. 随着 CPU 核数增加, 启动阶段时间甚至可能超过求解时间, 成为性能瓶颈. 尽管近年来从不同角度提出一些优化策略^[8~10], 最近的研究表明^[11], 随着 CPU 核数不断增加, 相对于求解阶段, 启动阶段的可扩展性越来越具有挑战性, 仍然是当前 AMG 算法大规模计算最主要的瓶颈之一.

现有 AMG 算法隐含有一个假设, 即: 求解对象是单个方程组, 而且该方程组是一个病态系统. 而在实际应用中, 需要求解的是一个包含很多方程组的序列. 受应用特征的影响, 一方面序列内每个方程的性质各不相同, 另一方面, 某些局部子序列的方程组具有相似性. 本文针对方程组序列的求解, 利用序列内方程组性质的差异和相似性, 设计了基于自适应启动策略的 AMG 预条件子. 该策略通过设计启动条件, 使得序列中的每个方程自适应地执行启动过程, 尽可能减少序列求解中的启动次数, 从而减少启动开销, 以提高并行可扩展性能. 惯性约束聚变 (ICF) 中辐射流体力学数值模拟的典型应用验证了该自适应策略的有效性.

本文安排如下: 第 2 节简单介绍 AMG 算法, 第 3 节介绍自适应 AMG 预条件子策略, 第 4 节给出在 ICF 数值模拟中的应用效果, 第 5 节总结.

2 AMG 算法

忽略序号, 考虑方程组序列 (1) 的单个方程 $Ax = b$. 记 $\Omega = \{x_1, x_2, \dots, x_n\}$ 为该方程对应的自由度集合, 并将其视为最细网格层. 以两层网格为例, 算法 1 给出了 AMG 求解该方程的流程^[3].

算法 1 AMG 算法 (两层网格情形)

1. 启动阶段 (setup): 构造多重网格算法组件.
 - 1.1 选取粗网格 Ω^c : 在细网格 Ω 中选取一个子集 $\Omega^c \subseteq \Omega$, 并将其视为粗网格;
 - 1.2 构造粗网格与细网格的层间插值算子 P 和限制算子 R : $P: \Omega^c \rightarrow \Omega$, $R: \Omega \rightarrow \Omega^c$
 - 1.3 构造粗网格矩阵: $A^c = RAP$ (Galerkin 方法);
 2. 求解阶段 (solve): 反复执行如下多重网格循环 (cycle), 直到收敛.
 - 2.1 前光滑: 在细网格层, 采用光滑子对 $Ax = b$ 执行 μ_1 次光滑迭代, 得到近似解 x^f ;
 - 2.2 计算残差并限制到粗网格: $b^c = R(b - Ax^f)$;
 - 2.3 求解粗网格方程: $A^c x^c = b^c$;
 - 2.4 插值并校正细网格近似解: $x^f = x^f + Px^c$;
 - 2.5 后光滑: 在细网格层, 采用光滑子对 $Ax = b$ 执行 μ_2 次光滑迭代, 更新近似解 x^f ;
-

表 1 AMG 并行可扩展性, 三维 Poisson 方程, 网格规模 15 亿
Table 1 Parallel scalability of AMG, 3D Poisson equation with 1.5 billion mesh size

Numbers of CPU cores	3000	6000	12000	24000
CPU time of setup phase (s)	17.6	8.4	6.9	15.5
CPU time of solve phase (s)	35.3	16.6	8.8	5.5
Total CPU time (s)	52.9	25.0	15.7	21.0

从算法 1 可知, AMG 算法通过构造一个比原矩阵 (细网格矩阵) 规模小的粗网格矩阵, 将粗网格的解校正到细网格上. 递归地执行此过程, 可进一步构造规模更小的粗网格, 直到某个网格层的规模足够小为止, 即可得到多层情形的 AMG 算法. 在求解阶段, 前后光滑通常采用 Jacobi 或近似块 Jacobi 迭代, 光滑次数 μ_1, μ_2 通常取为常数, 例如 1 或 2.

实际应用中, AMG 算法并不单独使用, 而是作为预条件技术来加速通用迭代方法的收敛速度^[12,13], 如 CG, GMRES, BiCGSTAB 等 Krylov 子空间方法^[14]. 预条件 Krylov 子空间方法的本质是将 Krylov 子空间方法求解原方程转为求解等价方程 $M^{-1}Ax = M^{-1}b$ (以左预条件为例), 其中 M^{-1} 称为预条件子. 一个好的预条件子需要尽可能逼近 A^{-1} , 同时要求构造简单. 常用的预条件子包括不完全 LU 分解 (ILU)^[15]、多重网格 (MG)^[16]、区域分解 (DDM)^[17] 等. 除了少数简单情形之外, 例如对角逆预条件子 $M^{-1} = D^{-1}$ (D 为 A 的对角部分), M^{-1} 的构造通常并不显式计算, 而是通过某种迭代实现. 对于 AMG 预条件子, M^{-1} 即是通过若干次多重网格循环 (cycle) 来实现. 因此, AMG 作为预条件子时, 对每个方程仍然需要执行算法 1 所示的启动和求解两个阶段.

AMG 算法的主要性能瓶颈在于启动阶段的构造. 表 1 给出了在某高性能计算机上 AMG 算法求解三维 Poisson 方程的并行可扩展性能测试结果. 该算例基于结构网格上的 7 点格式离散, 采用 AMG 算法单独求解, 固定 50 次迭代 (cycle). 测试时, 固定网格规模为 15 亿, CPU 核数从 3000 增加到 24000. 可以看到, 随着 CPU 核数增加, 启动阶段的可扩展性远不如求解阶段, 启动时间占总时间的比例由 3000 核的 33.3% 增加到 24000 核的 73.8%, 逐渐占主导, 成为性能瓶颈. 实际上, CPU 核数从 12000 增加到 24000 时, 启动阶段并没有获得加速.

导致启动阶段并行效率下降的主要原因是粗网格矩阵的构造. 如算法 1 所示, 粗网格矩阵采用 Galerkin 方法构造, 涉及 3 个稀疏矩阵的乘积运算 (算法 1 第 1.3 步, 记为 RAP). RAP 计算将改变各网格层矩阵的稀疏结构. 在表 1 的算例中, 网格层数为 12, 每个网格层均涉及 RAP 运算, 随着网格从最细层到最粗层, 矩阵变得越来越稠密、规模不断减小且稀疏结构越来越复杂, 导致粗网格层上 RAP 运算的通信模式不断复杂、通信与计算比例越来越大, 由此影响了大规模计算的并行可扩展性. 在文献 [8] 中, 作者详细分析了各网格层的通信模式.

3 自适应 AMG 预条件子

第 2 节的分析和测试表明, 降低启动阶段的开销是提高 AMG 并行可扩展性的关键. 对典型的非线性多物理耦合问题, 每个时间步涉及多个物理过程, 每个物理过程基于不同的 PDE 方程离散, 不同物理过程导出的矩阵规模和性质各不相同. 在同一物理过程, 物理现象随时间步进发生变化, 由于非线性的影响, 离散矩阵的性质也随之发生变化. 例如, 在 ICF 的辐射流体力学计算中, 一个典型的模拟需要数千至数万个时间步, 每个时间步涉及辐射扩散、电子和离子热传导过程的求解, 每个过程需进行非线性迭代, 一次完整模拟需要求解的方程组数目达数万至数百万, 这些方程的性质各不相同, 甚

至相差很大.

现有 AMG 预条件策略忽略了序列内方程组性质差异这一特征, 对所有方程均执行启动过程. 然而, 不同性质的方程应该采用不同的启动策略, 对某些方程组, 启动甚至是没有必要的. 例如, 对具有很强对角占优性质的系统, 基于简单迭代的光滑子作为预条件子即可快速收敛, 而无需启动过程, 但在现有策略中, 在迭代求解之前启动已经完成, 启动浪费已经造成.

基于上述分析, 本文提出自适应 AMG 预条件策略. 首先, 在待求解的方程组序列中, 我们把来源于同一个物理过程 (同一组 PDE) 的线性系统构成一个子序列, 称之为同类子序列. 对于同类子序列中的每个方程组, 其自适应 AMG 预条件策略按如下 3 个阶段执行:

1. **单层光滑.** 在最细网格层, 采用光滑子作为预条件子进行迭代, 直到满足启动条件, 转向第 2 步. 否则, 执行迭代直到收敛.

2. **多层重用.** 重用序列中前一次启动过程所构造的 AMG 预条件子进行迭代, 直到再次满足启动条件, 转向第 3 步. 否则, 执行迭代直到收敛.

3. **多层更新.** 基于当前矩阵构造新的 AMG 预条件子并保存, 采用新预条件子进行迭代, 直到收敛.

在上述自适应策略中, 首先, 启动阶段并不事先构造, 而是尽可能地发挥最细网格层光滑子的潜力, 尽量避免启动阶段的构造. 理想情况下, 如果达不到启动条件, 则意味着单层迭代即可收敛, 无需启动过程. 其次, 如果最细网格上的光滑子作为预条件子无法有效求解方程组, 则利用同类子序列方程组的相似性, 尽可能重用序列中前一个已经构造的预条件子, 尽量减少启动次数. 该策略的关键是启动条件的设计, 下面详细介绍.

3.1 启动条件

启动条件的设计原则是希望恰到好处地退出单层光滑阶段或多层重用阶段的预条件迭代求解. 我们提出一种基于当前阶段预条件子收敛行为的自适应启动条件, 即

$$\frac{\|r^i\|}{\|r^{i-1}\|} > \delta, \quad (2)$$

其中 $\|r^i\|$ 是当前阶段预条件子第 i 次迭代的残差范数, δ 是残差下降因子的允许上界. 在每个迭代步, 启动条件自动地监控当前预条件子的收敛行为, 如果当前迭代的残差下降量大于给定的允许上界 δ , 则进行预条件子的切换, 重用或更新 AMG 预条件的启动.

启动条件 (2) 的关键是确定允许上界 δ , 我们提出一种自适应计算方法. 记 ε 是迭代收敛阈值. 对当前预条件子, 假设最大允许迭代次数为 m , 当前处于第 i 次迭代, $\tau = \|r^i\| / \|r^0\|$ 是前 i 次迭代残差下降的总量, 则允许上界 δ 按如下计算:

$$\delta^{m-i} = \frac{\varepsilon}{\tau}. \quad (3)$$

上式表明, 允许上界即为能够在剩余 $m - i$ 次迭代达到收敛的残差下降平均因子, 反映了当前预条件子在允许的迭代次数内达到收敛的潜力. 给定 m , 随着迭代进行, 残差下降因子允许上界 δ 动态变化. 特别地, 对具有残差单调下降性质的迭代方法 (预条件 Krylov 子空间方法通常具有该性质), δ 会随着迭代的进行而逐渐放大, 使得启动条件变弱, 从而允许当前预条件子尽可能地多做有效迭代, 降低 AMG 启动的可能性, 达到降低启动次数的目的.

剩下的一个问题是需要确定当前预条件子的最大迭代次数 m . 这取决于 AMG 预条件子的启动代价, 为此, 引入两个参数 α 和 β , 分别刻画启动开销与单层光滑迭代开销之比, 以及启动开销与单次多层迭代 (单次 Cycle) 开销之比, 具体如下:

$$\alpha = \frac{T_{\text{setup}}}{T_{\text{sl}}}, \quad \beta = \frac{T_{\text{setup}}}{T_{\text{cycle}}}, \quad (4)$$

其中 T_{setup} 是 AMG 预条件的启动 CPU 时间, T_{sl} 和 T_{cycle} 分别是 AMG 求解阶段单层光滑和单次 Cycle 的 CPU 时间. 一方面, 对于容易求解的方程, 期望在前两个阶段即可成功求解, 另一方面, 对于前两个阶段无法成功求解的病态方程, 则希望尽快转到第 3 个阶段, 减少前两个阶段低效率迭代引入的冗余开销. 在此启发下, 将单层光滑阶段和多层重用阶段预条件的最大迭代次数 m^{sl} 和 m^{reuse} 分别设为

$$m^{\text{sl}} = \alpha, \quad m^{\text{reuse}} = \beta. \quad (5)$$

注意到, 启动条件 (2)~(5) 是一组启发式条件. 在式 (4) 中, 参数 α 和 β 定义为 CPU 时间比值. 对于给定的光滑子和 AMG 算法, α 和 β 仅与矩阵的稀疏模式和通信模式有关. 给定离散网格 (包括网格类型和规模)、网格剖分和 CPU 核数, 矩阵的稀疏模式和通信模式即可确定, 且在模拟过程保持不变. 因此, 这两个参数可在模拟初始时刻通过实际测试得到, 在模拟过程视为常数.

3.2 基于自适应启动的 AMG 预条件子 (α Setup-AMG)

基于上述启动条件的自适应 AMG 策略记为 α Setup-AMG. 记同类子序列矩阵为 $\{A_i, i = 1, \dots, n_{\text{eq}}\}$, n_{eq} 为该序列中方程数目, 算法 2 给出了 α Setup-AMG 的具体描述.

算法 2 自适应 AMG 预条件子 (α Setup-AMG)

初始化: $n_{\text{sl}} = 0, n_{\text{reuse}} = 0$, 当前最新的预条件子 $M^* = \{\text{NULL}\}$.

FOR($i = 1, 2, \dots, n_{\text{eq}}$) {

1. (单层光滑) 在最细网格层, 采用光滑子作为预条件子进行迭代, 直到收敛或满足启动条件.

1.1 如果收敛: $n_{\text{sl}} = n_{\text{sl}} + 1$, 转为下一个方程求解;

1.2 如果满足启动条件: 转为第 2 步;

2. (多层重用) 如果 $M^* \neq \text{NULL}$, 即上一个 AMG 预条件子存在, 则采用 M^* 作为 A_i 的预条件子, 直到收敛或满足启动条件.

2.1 如果收敛: $n_{\text{reuse}} = n_{\text{reuse}} + 1$, 转为下一个方程求解;

2.2 如果满足启动条件或 $M^* == \text{NULL}$, 转为第 3 步;

3. (多层更新) 构造 A_i 的 AMG 预条件子 M_i , 并存储于 $M^* = M_i$, 采用 M_i 作为预条件子, 直到收敛.

}

在算法 2 中, n_{sl} 和 n_{reuse} 分别为该序列在单层光滑阶段和多层重用阶段成功收敛的次数. 自适应预条件策略的目的是期望 n_{sl} 和 n_{reuse} 尽可能大, 这样实际启动次数就少. 具体地, 记 $R_{\text{sl}} = n_{\text{sl}}/n_{\text{eq}}$ 和 $R_{\text{re}} = n_{\text{reuse}}/n_{\text{eq}}$ 分别为该序列中单层光滑阶段和多层重用阶段的成功率, 则该序列求解中自适应策略导致启动次数减少的比例为

$$R_{\text{setup}} = R_{\text{sl}} + R_{\text{re}}, \quad (6)$$

R_{setup} 可用于评估自适应预条件策略的加速效果.

算法 2 的自适应 AMG 预条件策略已集成到 JASMIN 框架^[18]线性解法器中, 目前单层光滑子取为混合松弛迭代^[4] (进程内用 Gauss-Seidel, 进程间采用 Jacobi 迭代).

在算法 2 中, 启动条件作用于两个阶段, 其中第一阶段的启动条件目前仅针对最细网格层. 一个自然的想法是将其推广至多层情形, 即在每层的粗网格构造中采用启动条件, 由此 AMG 的层数也可以自适应地确定, 可以尽可能减少 AMG 的网格层数, 从而减小由粗网格带来的并行效率的损失. 这是下一步工作.

4 在 ICF 数值模拟中的应用

本节, 将本文提出的自适应 AMG 预条件子应用于惯性约束聚变 (ICF) 数值模拟^[19,20]. ICF 是一类典型的多物理耦合计算问题, 其核心是求解辐射流体力学方程组 (RHD). 该方程是一组流体力学与辐射能量运输组成的非线性偏微分方程组, 描述 ICF 中流体运动与能量传输过程. 在单群扩散近似下, 能量传输过程可通过如下时间相关的扩散方程组描述:

$$\frac{\partial E_r}{\partial t} = \nabla \cdot \left(\frac{c\lambda(E_r)}{\kappa_R} \nabla E_r \right) + \kappa_P(4\sigma T_e^4 - cE_r) + S_r, \quad (7)$$

$$\frac{\rho c_e \partial T_e}{\partial t} = \nabla \cdot (D_e(T_e) \nabla T_e) - \kappa_P(4\sigma T_e^4 - cE_r) + \omega_{ie}(T_i - T_e), \quad (8)$$

$$\frac{\rho c_i \partial T_i}{\partial t} = \nabla \cdot (D_i(T_i) \nabla T_i) - \omega_{ie}(T_i - T_e), \quad (9)$$

其中辐射能量密度 E_r 、电子温度 T_e 和离子温度 T_i 是上述方程待求解的物理量. 流体密度 ρ 在流体运动过程更新. 在方程 (7) 中, $\lambda(E_r)$ 是非线性限流因子, κ_P 和 κ_R 分别为 Planck 和 Rosseland 平均吸收系数, σ 是吸收平均自由程, S_r 是辐射源项, c 是光速. 在方程 (8) 和 (9) 中, $D_e(T_e)$ 和 $D_i(T_i)$ 分别为电子和离子非线性热传导系数, c_e 和 c_i 分别是电子和离子的比热, ω_{ie} 是电子与离子的能量交换系数.

由于辐射能量密度 E_r 与辐射温度 T_r 具有关系 $E_r = aT_r^4$, 因此, 耦合方程 (7)~(9) 也称为三温方程. 时间方向隐式离散后, 由于该方程是非线性方程, 在 ICF 程序中, 在每个时间步, 通过 Picard 迭代或 Newton 迭代进行线性化^[21], 每个非线性迭代求解一个线性代数方程组, 随着时间步推进, 需要求解一个方程组序列. 针对三温方程, 现有工作主要集中于改善 AMG 算法的收敛性^[22~25], 没有考虑启动阶段的并行可扩展性.

4.1 测试模型

所有测试基于三维辐射流体界面不稳定性程序 LARED-S^[26] 进行¹⁾, 该程序是一个基于 Euler 方法的结构网格程序, 用于模拟辐射驱动下靶丸内爆过程中流体界面不稳定性现象^[27]. 在该程序中, 三温耦合方程 (7)~(9) 采用算子分裂方法进行求解, 每个 Picard 非线性迭代需要分别求解辐射、电子和离子方程, 形成 3 个待求解的同类方程组序列.

测试模型是某个内爆过程减速阶段界面不稳定性模拟^[28], 网格规模 $500 \times 500 \times 500$ (1.25 亿), 全过程需 2778 个时间步, 分别求解 12879 个辐射、电子和离子方程, 形成 3 个方程组序列.

我们将 α Setup-AMG 与传统 AMG 预条件子进行对比, 其中传统 AMG 预条件子采用目前国际上广泛使用的 Hypr²⁾ 预条件库中的 BoomerAMG 解法器^[29], 这是目前国际上公认最好的 AMG 解

1) LARED-S 程序: <http://www.iapcm.ac.cn/jasmin/index.php?page=lareds>.

2) Hypr software: High performance preconditioners, <http://www.llnl.gov/CASC/hypr/>.

表 2 两个算法性能比较: ICF 三维三温模型, 每个序列方程组数目 12879, 3125 个 CPU 核

Table 2 Performance comparison: 3D-3T model of ICF, $n_{eq} = 12879$, 3125 cores

	Overall simulation	AMG solver part		
		Setup phase	Solve phase	Setup+solve
CPU time of BoomerAMG (h)	19.7	8.3	7.1	15.4
CPU time of α Setup-AMG (h)	14.5	4.2	6.0	10.2
Improvement of α Setup-AMG (%)	26.4	49.4	15.5	33.8

表 3 α Setup-AMG 策略统计信息: ICF 三维三温模型, 每个序列方程组数目 12879Table 3 Statistic information of α Setup-AMG: 3D-3T model of ICF, $n_{eq} = 12879$

	R_{sl} (%)	R_{re} (%)	R_{setup} (%)
Radiation equation series	3	9	12
Electron equations series	6	18	24
Ion equation series	10	63	73
Average	9	30	39

法器之一. 测试中, BoomerAMG 和 α Setup-AMG 均作为 GMRES(20) 的预条件子. 除了启动阶段之外, 两个算法的其他所有参数, 包括粗化策略、插值算子、光滑子等都取为相同, 其中, 光滑子采用目前实际应用中普遍使用的前后 1 次混合松弛迭代. 经测试, 对该模型, α Setup-AMG 预条件子中启动条件的两个参数分别取为 $\alpha=20$, $\beta=5$.

4.2 测试结果

所有测试在某高性能计算机上进行. 首先, 测试全过程模拟的总体计算效率, 采用 3125 个 CPU 核, 表 2 给出测试结果. 结果表明, 相对于 BoomerAMG, 自适应策略 α Setup-AMG 将全过程模拟时间从 19.7 h 降为 14.5 h, 性能提升 26.4%, 其中解法器总时间从 15.4 h 降到 10.2 h, 性能提升 33.8%. 可以看到, 计算效率的提高主要来源于启动阶段, 该阶段 CPU 时间从 8.3 h 下降到 4.2 h, 性能提升 49.4%, 表明 α Setup-AMG 显著降低了启动开销.

根据 α Setup-AMG 的算法流程, 自适应启动策略的加速效果可以通过求解序列中启动次数减少比例 R_{setup} 来衡量, 包括单层光滑阶段预条件迭代成功率 R_{sl} 和多层重用阶段预条件迭代成功率 R_{re} 两个方面, 见式 (6). 具体地, 针对模拟过程的 3 个方程组序列, 表 3 给出了 α Setup-AMG 策略中这些参数的统计.

从表 3 可知, 对不同的方程组序列, α Setup-AMG 策略的性能行为各不相同, 即: 前两个阶段的收敛成功率和启动次数下降比例均不相同, 这反映了在多物理计算中, 不同物理过程之间方程性质的差异. 对辐射、电子、离子方程, α Setup-AMG 策略启动次数减少比例分别为 12%, 24% 和 73%, 这表明, 这 3 个方程序列的求解难度依次增加. 总体上, 3 个序列平均减少启动次数 39%. 比较表 2 的测试结果, 与 BoomerAMG 预条件相比, α Setup-AMG 在 3125 个 CPU 核上将启动时间减少了 49.4%, 高于表 3 中的启动次数平均减少比例 39%, 这隐含着 α Setup-AMG 具有更好的并行可扩展性.

具体地, CPU 核数从 625 增加到 6250, 表 4 给出了前 100 个时间步两个算法的解法器并行效率测试结果. 可以看到, 相对于 625 个 CPU 核, 在 3125 核和 6250 核上, α Setup-AMG 将并行效率分别从 47% 和 25% 提高到 61% 和 41%, 解法器时间分别减少 35.2% 和 47.5%, 表明 α Setup-AMG 对并行

表 4 两个算法并行效率比较: ICF 三维三温模型 (100 个时间步)

Table 4 Parallel efficiency comparison: 3D-3T of ICF (100 steps)

	Numbers of CPU cores	625	1250	3125	6250
BoomerAMG	CPU time (min)	80.1	45.0	34.1	31.4
	Parallel efficiency (%)	100	89	47	25
	CPU time (min)	67.9	37.3	22.1	16.5
α Setup-AMG	Parallel efficiency (%)	100	91	61	41

可扩展性能有明显改善.

5 总结

AMG 预条件技术普遍应用于很多实际数值模拟中, 启动阶段是其固有的不可扩展因素, 严重影响大规模计算下并行可扩展性, 随着 CPU 核数增加, AMG 并行效率显著下降. 本文针对实际应用中方程组序列的求解, 根据序列中方程性质的差异性和相似性, 提出基于自适应启动策略的 AMG 预条件 α Setup-AMG, 其目的是尽可能减少启动次数, 提高并行可扩展性. 应用于 ICF 数值模拟的实际模型计算表明了 α Setup-AMG 的有效性.

现有的通用迭代方法通常具有较高的通信与计算比^[30], 随着高性能计算机规模不断增加, 迭代方法的通信开销越来越成为性能瓶颈, 预条件子为缓解这一瓶颈提供了新的机会, 其中利用应用特征充分挖掘现有预条件算法的潜力是一个重要问题. 本文的动机主要针对 AMG 算法, 但所提出的自适应策略是一个具有普适性的预条件自适应调优策略. 基于该策略, 下一步我们将尝试在多个预条件子之间进行多级自适应, 尽可能减小在大规模计算下现有预条件算法中固有的不可扩展因素对数值模拟并行性能的影响.

致谢 本文数值模拟应用测试是在北京应用物理与计算数学研究所激光聚变研究室开发的 ICF 内爆不稳定性模拟程序 LARED-S^[26] 上完成的, 在测试过程中, 程序研发团队成员范征锋研究员等为本文提供了测试模型, 在此向他们表示感谢.

参考文献

- 1 Brandt A, McCormick S, Ruge J. Algebraic multigrid for sparse matrix equations. In: Sparsity and its Application. Cambridge: Cambridge University Press, 1984. 257–284
- 2 Ruge J W, Stueben K. Algebraic multigrid. In: Multigrid Methods. Philadelphia: SIAM, 1987. 73–130
- 3 Stueben K. Algebraic Multigrid (AMG): An Introduction With Applications. Sankt Augustin: GMD-Forschungszentrum Informationstechnik, 1999
- 4 Yang U M. Parallel algebraic multigrid methods—high performance preconditioners. In: Numerical Solution of Partial Differential Equations on Parallel Computers. Berlin: Springer-Verlag, 2006. 209–236
- 5 Xu X W, Mo Z Y. Scalability analysis for parallel algebraic multigrid algorithms. Chinese J Comput Phys, 2007, 24: 387–394 [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: 387–394]
- 6 Mo Z Y, Xu X W. Relaxed RS0 or CLJP coarsening strategy for parallel AMG methods. Parall Comput, 2007, 33: 174–185
- 7 Baker A H, Gamblin T, Schulz M, et al. Challenges of scaling AMG across modern multicore architectures. In: Proceedings of the 25th IEEE International Symposium on Parallel and Distributed Processing (IPDPS), Anchorage, 2011. 275–286

- 8 Baker A H, Falgout R D, Gamblin T, et al. Scaling AMG solvers: on the road to Exascale. In: Proceedings of International Conference on Competence in High Performance Computing, Schwetzingen, 2010. 215–226
- 9 Baker A H, Falgout R D, Gahvari H, et al. Preparing Algebraic Multigrid for Exascale. LLNL Technical Report LLNL-TR-533076. 2012
- 10 Falgout R, Schroder J. Non-Galerkin coarse grids for algebraic multigrid. *SIAM J Sci Comput*, 2014, 36: 309–334
- 11 Park J, Smelyanskiy M, Yang U M, et al. High-performance algebraic multigrid solver optimized for multi-core based distributed parallel systems. In: Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis (SC'15). New York: ACM, 2015. 54
- 12 Benzi M. Preconditioning techniques for large linear systems: a survey. *J Comput Phys*, 2002, 182: 418–477
- 13 Gu T X, Xu X W, Liu X P, et al. Iterative Methods and Preconditioning Techniques. Beijing: Science Press, 2015 [谷同祥, 徐小文, 刘兴平, 等. 迭代方法和预处理技术 (下册). 北京: 科学出版社, 2015]
- 14 Saad Y. Iterative Methods for Sparse Linear Systems. 2nd ed. Philadelphia: Society for Industry and Applied Mathematics, 2003
- 15 Meijerink J A, van der Vorst H A. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math Comput*, 1977, 31: 148–162
- 16 Trottenberg U, Oosterlee C W, Schuller A. Multigrid. London: Academic Press, 2001
- 17 Toselli A, Widlund O. Domain Decomposition Methods: Algorithms and Theory. Philadelphia: Society for Industry and Applied Mathematics, 2004
- 18 Mo Z Y, Zhang A Q, Cao X L, et al. JASMIN: a parallel software infrastructure for scientific computing. *Front Comput Sci China*, 2010, 4: 480–488
- 19 Pei W B. The construction of simulation algorithm for laser fusion. *Commun Comput Phys*, 2007, 2: 255–270
- 20 Pei W B, Zhu S P. Scientific computing for laser fusion. *Physics*, 2009, 38: 559–568 [裴文兵, 朱少平. 激光聚变中的科学计算. *物理*, 2009, 38: 559–568]
- 21 An H B, Mo Z Y, Xu X W, et al. On choosing a nonlinear initial iterate for solving the 2-D 3-T heat conduction equations. *J Comput Phys*, 2009, 228: 3268–3287
- 22 Baldwin C, Brown P, Falgout R, et al. Iterative linear solvers in 2D radiation-hydrodynamics code: methods and performance. *J Comput Phys*, 1999, 154: 1–40
- 23 Xu X W, Mo Z Y, An H B. Algebraic two-level iterative method for 2D-3T radiation diffusion equations. *Chinese J Comput Phys*, 2009, 26: 1–8 [徐小文, 莫则尧, 安恒斌. 求解二维三温辐射扩散方程组的一种代数两层网格迭代方法. *计算物理*, 2009, 26: 1–8]
- 24 Zhou Z Y, Xu X W, Shu S, et al. An adaptive two-level preconditioner for 2D-3T radiation diffusion equations. *Chinese J Comput Phys*, 2012, 29: 475–483 [周志阳, 徐小文, 舒适, 等. 二维三温辐射扩散方程两层预条件子的自适应求解. *计算物理*, 2012, 29: 475–483]
- 25 Yue X Q, Shu S, Xu X W, et al. An adaptive combined preconditioner with applications in radiation diffusion equations. *Commun Comput Phys*, 2015, 18: 1313–1335
- 26 Fan Z F, Xu X W, Sun W J, et al. Radiation hydrodynamic instabilities simulation code LARED-S. In: Proceedings of the 16th National Conference on Numerical Methods for Hydrodynamic, Kunming, 2013. 25–26 [范征锋, 徐小文, 孙文俊, 等. 辐射流体界面不稳定性模拟程序 LARED-S. 见: 第十六届全国流体力学数值方法研讨会论文集, 昆明, 2013. 25–26]
- 27 Zhang W Y, Ye W H, Wu J F, et al. Hydrodynamic instabilities of laser indirect-drive inertial-confinement-fusion implosion. *Sci Sin-Phys Mech Astron*, 2014, 44: 1–23 [张维岩, 叶文华, 吴俊峰, 等. 激光间接驱动聚变内爆流体不稳定性研究. *中国科学: 物理学 力学 天文学*, 2014, 44: 1–23]
- 28 Fan Z F, Zhu S P, Pei W B, et al. Numerical investigation on the stabilization of the deceleration phase Rayleigh-Taylor instability due to alpha particle heating in ignition target. *Euro Phys Lett*, 2012, 99: 5003
- 29 Henson V E, Yang U M. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Appl Numer Math*, 2002, 41: 155–177
- 30 Xu X W, Mo Z Y, Wu L P. Analysis of communication-to-computation based on asymptotic size for iterative methods. *Chinese J Comput*, 2013, 36: 782–789 [徐小文, 莫则尧, 武林平. 迭代方法中基于渐近规模的通信与计算比分析. *计算机学报*, 2013, 36: 782–789]

An adaptive AMG preconditioning strategy for solving large-scale sparse linear systems

Xiaowen XU^{1,2*}, Zeyao MO^{1,2} & Hengbin AN^{1,2}

1 Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing 100094, China;

2 CAEP Software Center for High Performance Numerical Simulation, Beijing 100086, China

*E-mail: xwxu@iapcm.ac.cn

Abstract A series of sparse linear systems must be solved in applications that are based on the implicit solution of time-dependent partial differential equations (PDEs). Preconditioned iterative methods are usually employed to solve such sparse linear systems. AMG is one of the most popular preconditioners in real applications. However, it results in poor parallel scalability, owing to its setup phase. In this paper, by utilizing the differences and similarities in property among the systems in series, an adaptive AMG preconditioning strategy is presented to improve the parallel scalability. The results obtained for a radiation hydrodynamics computation within an ICF simulation demonstrate the efficiency and improvement of the adaptive strategy. For a typical model, the new strategy improves the parallel efficiency from 47% to 61%, and reduces the CPU time from 19.7 h to 14.5 h.

Keywords sparse linear solver, iterative methods, preconditioner, algebraic multigrid (AMG), parallel computing



Xiaowen XU was born in 1978. He received his B.S degree in computational mathematics from Xiangtan University (2002), and his Ph.D. degree in computational mathematics from the Graduate School of CAEP (2007), both in China. Currently, he is a professor of IAPCM. His research interests include parallel algorithms, scalable linear solvers, and large-scale simulations for real applications.



Zeyao MO was born in 1971. He received his B.S in applied mathematics (1992) and Ph.D. in computer science (1997) both from National Defense University of Technology in China. Currently, he is a professor of IAPCM. His research interests include parallel computing, and parallel programming frameworks for large-scale numerical simulations.