

Rainmaker's Notebook

『求雨巫师的神奇之处在于他总是躲着不见你，却总说刚下完的雨是拜他所赐。』——《天真的人类学家》

Home

Archives

About

SiteXC

代数多重网格 (Algebraic Multigrid) 简介

代数多重网格 (AMG) 是利用几何多重网格 (Geometric Multigrid, GMG) 的一些重要原则和理念发展起来的不依赖于实际几何网格的多重网格方法。它继承了几何多重网格的主要优点，并且可以被用于更多类型的线性方程组。本文将介绍 Classic AMG 的基本算法并忽略数学推导过程。

AMG 和 GMG 一样，使用 smoothing 和 coarse-grid correction 这两个步骤来更新方程组的解。求解 $Ax = b$ 一个最简单的两层方法流程如下：

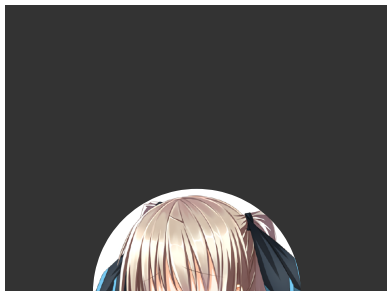
1. 前光滑：使用给定的 x_0 作为初始解执行 v_1 步迭代法求解 $Ax = b$ ，得到 $x^{(1)}$ ；
2. 细网格残差限制投影：构造 restriction operator R ，将残差 $r^{(1)} := b - Ax^{(1)}$ 投影到粗网格上： $r^C := Rr^{(1)}$ ；
3. 粗网格求解误差方程：构造粗网格系数矩阵 A^C ，求解 $A^C e^C = r^C$ ，其中 e^C 的初始猜测解为 0；
4. 粗网格误差插值投影：构造 prolongation operator P ，将粗网格求解得到的误差 e^C 投影到细网格上并校正 $x: x^{(2)} := x^{(1)} + Pe^C$ ；
5. 后光滑：使用 $x^{(2)}$ 作为初始解，执行 v_2 步迭代法求解 $Ax = b$ ，得到 x_{new} 。

AMG 和 GMG 的区别在于构造 R 和 A^C 。由于 AMG 不依赖于任何实际几何网格信息，所以必须通过其他方式来定义粗网格和构造 R ，并且有 $P = R^T$ 。同时，一般使用 Galerkin operator 作为粗网格： $A^C := RAR^T$ 。AMG 将系数矩阵视作有向邻接表，将这个表对应的图作为细网格（当前层），并在其上定义『粗网格』。下面我们只讨论 Classic AMG 的处理方法。

C-AMG 通过如下算法来构造粗网格：

1. 为图上每一个点赋权重，权重值等于其邻接的点的个数；
2. 选取当前图上未被分类的点中权重最大的点，将其加入粗网格格点集，将其所有邻居加入细网格格点集；
3. 对于所有新加入细网格格点集的点，将它们的邻居的权重 +1；
4. 重复 2-3，直至所有点都被分配完毕。

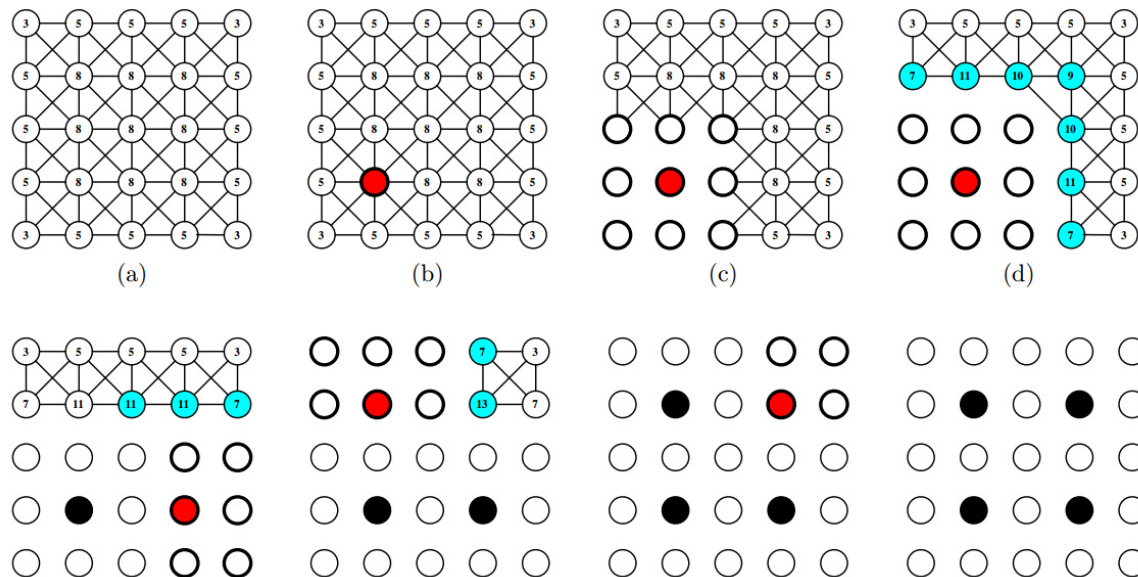
下图是来自 Ref. 2 一个二维网格上使用 9 点 stencil 的网格用 C-AMG 构造粗网格的样例。圆圈中是网格权重，红色表示每次选中新加入粗网格格点集的格点，黑色粗空心圈表示新加入细网格格点集的格



Rainmaker's Notebook

『求雨巫师的神奇之处在于他总是躲着不见你，却总说刚下完的雨是拜他所赐。』——《天真的人类学家》

点，蓝色表示新加入细网格格点集的点的邻居的权重更新。



C-AMG 中插值和限制投影算子的定义依赖于边权大小。C-AMG 首先使用这个定义：

强依赖：给定阈值 $0 < \theta \leq 1$ ，我们称变量 u_i 强依赖于 u_j ，如果满足以下关系：

$$-a_{ij} \geq \theta \cdot \max_{k \neq i} -a_{ik}.$$

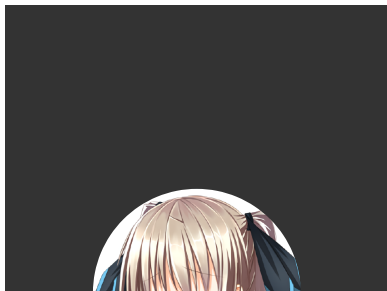
这里有两点值得注意：

1. 根据定义，大于 0 的非对角元素会被忽略；
2. 即使 A 是对称的，也有可能出现 u_i 强依赖于 u_j 但 u_j 不是强依赖于 u_i 。

对于每一个格点 u_i ，我们马上可以定义三个集合：

- C_i : 所有 u_i 强依赖的粗网格格点
- F_i^S : 所有 u_i 强依赖的细网格格点
- F_i^W : 所有 u_i 非强依赖的细网格格点

为了方便，我们讨论如何构造插值算子 P 。对于 u_i ，如果它本身是粗网格格点，那么不需要进行插值；否则它使用 C_i 中的粗网格格点值进行插值修正（即 $(Pe^C)_i$ 依赖于 $(e^C)_j, j \in C_i$ ）。插值修正有四步。



Rainmaker's Notebook

『求雨巫师的神奇之处在于他总是躲着不见你，却总说刚下完的雨是拜他所赐。』——《天真的人类学家》

[Home](#)

[Archives](#)

[About](#)

[SiteXC](#)

第一步，把 F_i^W 的边权值『重新分配』给 a_{ii} : $a_{ii}^{new} := a_{ii} + \sum_{k \in F_i^W} a_{ik}$. 下面的 Figure 5 是一个示例，左右两列的 F_i^W 格点与中间的 u_i 的边权被累加到了 u_i 上。

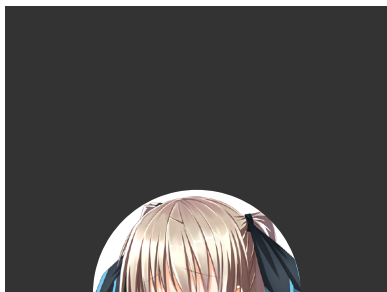
第二步，对于 F_i^S 中的每一个点 u_k ，统计它对相邻的 $u_l, l \in C_i$ 点的总『贡献』: $\delta_k := \sum_{l \in C_i} a_{kl}$.

第三步，对于 C_i 中的每一个点 u_j ，将与之相邻的 F_i^S 的边权『重新分配』给 u_j : $a_{ij}^{new} := a_{ij} + \sum_{k \in F_i^S} a_{kj} \frac{a_{ik}}{\delta_k}$. 或者说，将 F_i^S 中每一个点的边权『重新分配』给与之相邻的 C_i 中的点。

这里值得注意的是，如果 F_i^S 中的某个点 u_k 与 u_j 不相邻，则右端的 $a_{kj} = 0$.

第四步，插值修正: $s_i := \sum_{j \in C_i} w_{ij} s_j, w_{ij} := -\frac{a_{ij}^{new}}{a_{ii}^{new}}$.

藉此，我们便成功定义了插值算子 P . 下面是来自 Ref. 1 的一个示例:



Rainmaker's Notebook

『求雨巫师的神奇之处在于他总是躲着不见你，却总说刚下完的雨是拜他所赐。』——《天真的人类学家》

Home

Archives

About

SiteXC

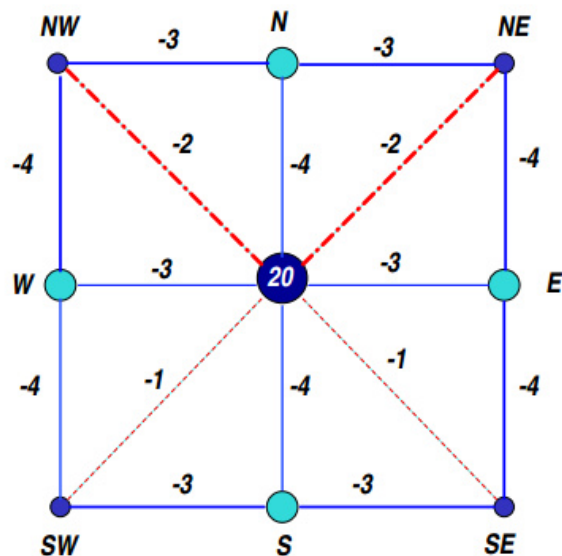


Figure 13.10: Darker filled circles represent the fine nodes. Thick dash-dot lines represent the strong connections. Values on edges are the a_{ij} 's. The value 20 at the center (fine) point is a_{ii} .

In this case, Equation (13.73) yields,

$$\begin{aligned} s_i &= \frac{1}{18} \left[4s_S + 4s_N + 3s_W + 3s_E + 2\frac{3s_N + 4s_W}{7} + 2\frac{3s_N + 4s_E}{7} \right] \\ &= \frac{1}{18} \left[4s_S + \left(4 + \frac{12}{7}\right) s_N + \left(3 + \frac{8}{7}\right) s_W + \left(3 + \frac{8}{7}\right) s_E \right]. \end{aligned}$$

Notice that, as is expected from an interpolation formula, the weights are all nonnegative and they add up to one. \square

图中中心点为 u_i , 待插值。N, W, S, E 四个点是粗网格点, NE, NW 是 u_i 强依赖的细网格点, SE, SW 是 u_i 非强依赖的细网格点, 边上各数值为边权。上面的公式中:

- 公分母 $\frac{1}{18} := \frac{1}{20-1-1}$, 即 $a_{mid}^{new} := a_{mid} + a_{mid-SW} + a_{mid-SE}$;
- 第一行最后两项的两个 7 分别是 $\delta_{NW} := a_{N-NW} + a_{NW-W}$ 和 $\delta_{NE} := a_{E-NE} + a_{NE-N}$;
- 第二行 s_S, s_N, s_W, s_E 前面的系数即为经过第三步边权重新分配以后新的边权。

这里另外有两个来自 Ref. 2 的计算插值系数的样例。Figure 4 中没有第一步，Figure 5 中没有第二、三步。这三个样例我都用来测试过我的样例代码。

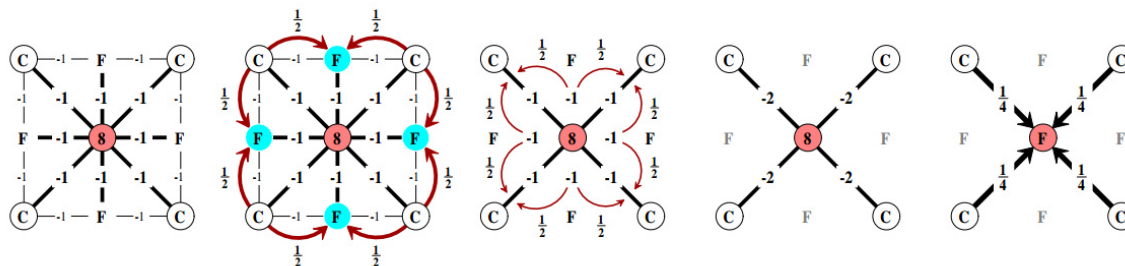


Figure 4: Derivation of C-AMG interpolation for the standard 9-point finite element stencil. In the second image, **we assume that strongly-connected F -points are interpolated from neighboring interpolatory points.** The weights (all $1/2$) are chosen based on the underlying matrix entries such that the constant function is interpolated exactly. **In the third image, we “redistribute” the strong F connections according to the interpolation weights in the previous step.** This produces the “collapsed stencil” in the fourth image, which leads directly to the interpolation rule in the last image.

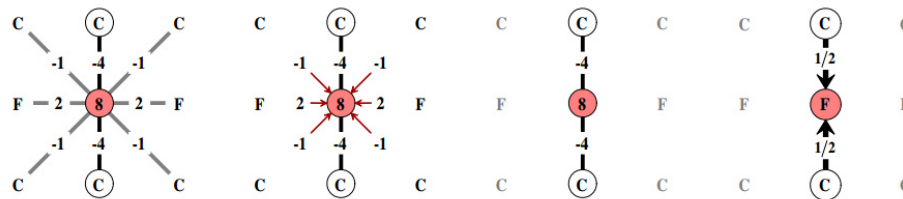
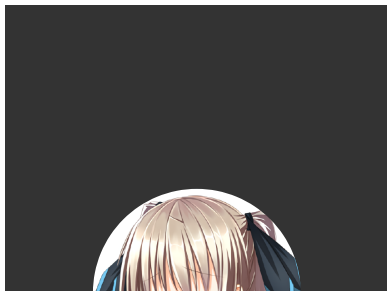


Figure 5: Derivation of C-AMG interpolation for an anisotropic 9-point finite element stencil. In the second image, **weak coefficients are added to the diagonal to produce the “collapsed stencil”** in the third image, which leads directly to the interpolation rule in the last image

样例代码

为了确保我的理解是正确的，我自己动手写了一个代码，[在这里](#)。我用它和之前写的 Geometric Multigrid ([代码](#), [文章](#)) 进行了对比，测试表明这两者的收敛行为非常接近，也证明我的代码和理解没有错。



Rainmaker's Notebook

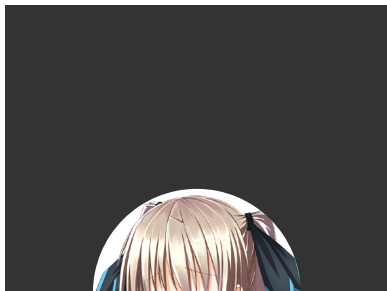
『求雨巫师的神奇之处在于他总是躲着不见你，却总说刚下完的雨是拜他所赐。』——《天真的人类学家》

[Home](#)

[Archives](#)

[About](#)

[SiteXC](#)



Rainmaker's Notebook

『求雨巫师的神奇之处在于他总是躲着不见你，却总说刚下完的雨是拜他所赐。』——《天真的人类学家》

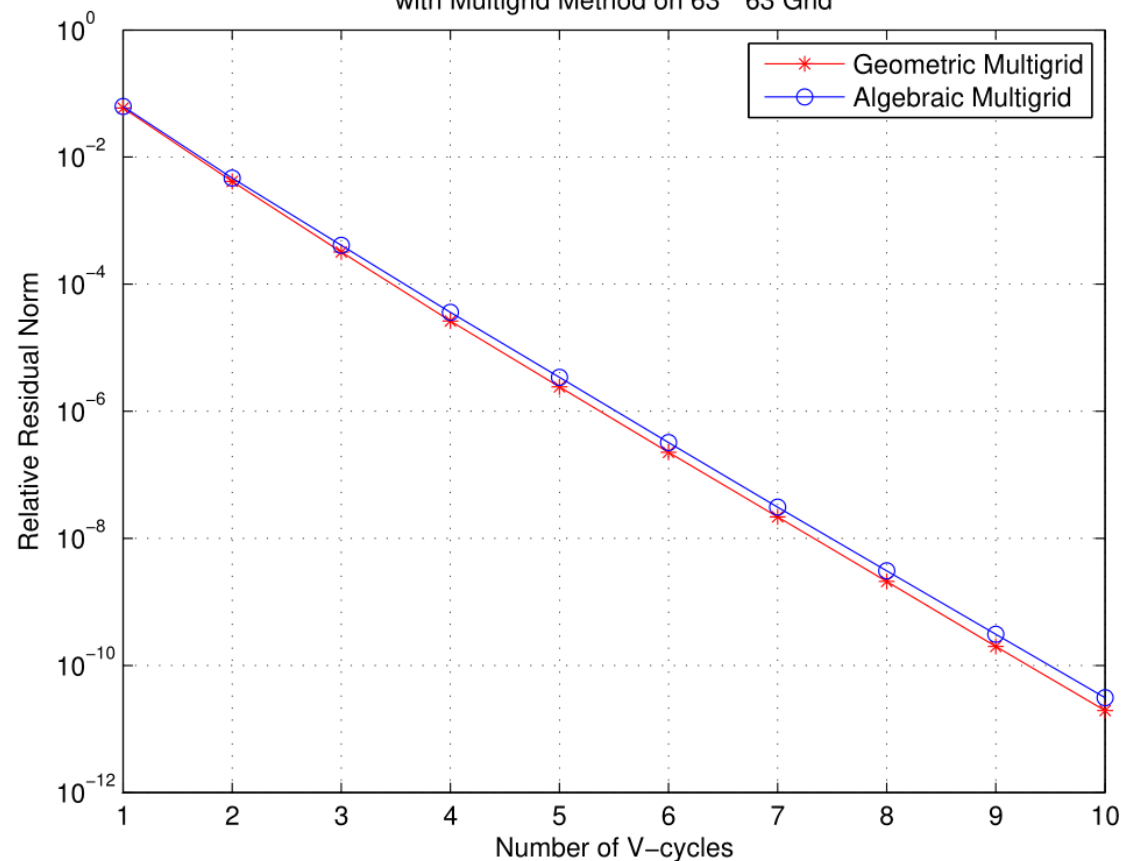
[Home](#)

[Archives](#)

[About](#)

[SiteXC](#)

Solving Center Difference Discreted Poisson Equation with Multigrid Method on 63 * 63 Grid



Reference

1. Yousef Saad, *Iterative Methods for Sparse Linear System (Second Edition)*, Philadelphia: SIAM, 2003 [Online Access](#)
2. R. D. Falgout, *An Introduction to Algebraic Multigrid*, in *Computing in Science and Engineering Journal*, Vol 8 Issue 6, 2006 [Online Access](#)

发布于 2018-01-01

/ tags: { [Multigrid](#) } { [LinearAlgebra](#) }

Related [Issues](#) not found