

等几何分析的多重网格共轭梯度法*

刘石¹, 陈德祥², 冯永新¹, 徐自力², 郑李坤¹

(1. 广东电网公司 电力科学研究院, 广州 510080;

2. 西安交通大学 航天航空学院; 机械结构强度与振动国家重点实验室, 西安 710049)

摘要: 提高 NURBS 基函数阶数可以提高等几何分析的精度, 同时也会降低多重网格迭代收敛速度. 将共轭梯度法与多重网格方法相结合, 提出了一种提高收敛速度的方法, 该方法用共轭梯度法作为基础迭代算法, 用多重网格进行预处理. 对 Poisson (泊松) 方程分别用多重网格方法和多重网格共轭梯度法进行了求解, 计算结果表明: 等几何分析中采用高阶 NURBS 基函数处理三维问题时, 多重网格共轭梯度法比多重网格法的收敛速度更快.

关键词: 等几何分析; 多重网格; 共轭梯度法; Poisson 方程; 迭代算法; NURBS

中图分类号: O241.82 **文献标志码:** A

doi: 10.3879/j.issn.1000-0887.2014.06.005

引 言

等几何分析是求解偏微分方程的一种新数值方法^[1], 它与有限元法一样都基于 Galerkin 变分, 但是采用非均匀有理 B 样条 (NURBS) 作为基函数. 因为 NURBS 基函数在单元边界上能够实现 $C^k (k \geq 1)$ 连续性, 提高光滑性 k 可以提高数值解的精度^[2-3], 所以等几何分析在精度方面优于传统有限元方法^[4].

当等几何分析产生的代数方程规模很大时, 直接求解计算成本过高, 一般采用迭代方法求解. 多重网格迭代是一种高效的迭代算法, 该方法在有限容积法^[5-6]、有限差分法^[7-8]和有限元方法^[9]中的研究较多, 但在等几何分析中的研究较少, 仅 Gahalaut 等^[10]在 2013 年研究了用多重网格算法求解等几何分析中的代数方程. 文献^[10]的结果表明多重网格方法具有很快的收敛速度, 但是当基函数的次数或光滑性增加时, 多重网格的收敛速度降低. 因此, 在提高基函数光滑性获得更高精度的同时, 多重网格的效率也随之降低.

多重网格的基本思想是在密网格上进行光滑, 使高频误差衰减, 在粗网格上进行误差修正, 使低频误差衰减^[7]. 常用的光滑算法是 Jacobi 迭代或 Gauss-Seidel 迭代, 随着基函数光滑性升高, Jacobi 迭代或 Gauss-Seidel 迭代的光滑效果降低, 多重网格的收敛速度降低. 共轭梯度法是一种 Krylov 子空间迭代方法, 它适用于系数矩阵为对称正定的情况. 理论上若代数方程的维数为 n , 那么进行 n 次共轭梯度法迭代将收敛到精确解; 实际计算中共轭梯度法的收敛速度快慢取决于所采用的预处理矩阵能否聚集系数矩阵的特征值^[11]. 多重网格方法可看成用简单迭

* 收稿日期: 2013-12-03; 修订日期: 2014-05-04

基金项目: 国家重点基础研究发展计划(973 计划)(2011CB706505); 国家自然科学基金(51275385)

作者简介: 刘石(1974—), 男, 湖北大冶人, 高级工程师, 博士 (E-mail: 13925041516@139.com);

陈德祥(1979—), 男, 合肥人, 博士生(通讯作者. E-mail: cdx97@tom.com).

代方法求解某个预处理矩阵作用下的代数方程,这个预处理矩阵能够有效聚集矩阵特征值,用多重网格预处理能够提高共轭梯度法的收敛速度^[12].本文提出了一种适用于等几何分析的多重网格共轭梯度法,当 k 较高时,那些在多重网格中衰减较慢的误差可在共轭梯度法中快速衰减.

本文用等几何分析对二维和三维 Poisson 方程进行了计算,分别用多重网格法和多重网格共轭梯度法来求解线性方程组,结果表明 NURBS 基函数阶数增加时多重网格共轭梯度法的收敛速度更快.

1 多重网格法

多重网格方法需要使用疏密不同的多组网格,等几何分析的疏密网格在计算中通过节点插入自动生成^[2].多重网格方法还需要在不同网格之间进行转换,文献[10]使用了网格转换矩阵,但没有给出计算关系式,本文根据离散 B 样条首次给出了等几何多重网格方法的转换矩阵关系式.

1.1 网格转换矩阵

对一维情况,给定两组节点向量 ξ_i^H 和 ξ_j^h ,上标 H,h 分别表示与疏、密网格相关的量, ξ_j^h 是由 ξ_i^H 插入节点得到的.由这两组节点向量定义的 B 样条基函数 $\{N_i^H\}_{i=1}^{n_h}$ 和 $\{N_j^h\}_{j=1}^{n_h}$ 满足如下线性变换关系^[13]:

$$N_i^H = \sum_{j=1}^{n_h} R_{i,j}^k N_j^h, \tag{1}$$

式中 k 为 B 样条的阶数. $R_{i,j}^r (r = 1, 2, \dots, k)$ 对 r 满足如下递归关系:

$$R_{i,j}^r = \omega_{i,j}^r R_{i,j}^{r-1} + (1 - \omega_{i+1,j}^r) R_{i+1,j}^{r-1}, \tag{2}$$

其中

$$\omega_{i,j}^r = \begin{cases} \frac{\xi_{j+r-1}^h - \xi_i^H}{\xi_{i+r}^H - \xi_i^H}, & \xi_i^H \neq \xi_{i+r}^H, \\ 0, & \text{other,} \end{cases} \quad R_{i,j}^1 = \begin{cases} 1, & \xi_i^H \leq \xi_j^h < \xi_{i+1}^H, \\ 0, & \text{other.} \end{cases}$$

$R_{i,j}^k$ 的递归关系与 B 样条基函数的定义类似,它的第 i 行是基函数 N_i^H 在点 $\{\xi_j^h\}_{j=k}^{n_h+k-1}$ 的函数值,因此又称为离散 B 样条.因为 N_i^H 在区间 $[\xi_i^H, \xi_{i+k}^H)$ 上大于 0,在其它点上等于 0,所以 $R_{i,j}^k$ 为稀疏矩阵.等几何分析的有限元空间从 B 样条或 NURBS 基函数导出,疏密网格下它们之间的转换关系与 B 样条基函数一样.取 $P = R^T$,则 P 和 R 分别是疏密网格之间的插值矩阵和限制矩阵,本文采用文献[13]中的算法来生成矩阵 R ,该算法充分利用了 R 的稀疏性来提高计算效率.

根据一维转换矩阵可以导出二维和三维转换矩阵.以二维为例,假设疏网格的节点向量为 ξ_i^H 和 η_j^H ,它们所定义的 B 样条基函数为 $\{N_i^H(\xi)\}_{i=1}^n$ 和 $\{N_j^H(\eta)\}_{j=1}^m$;插入节点后得到密网格的节点向量为 ξ_i^h 和 η_j^h ,相应的基函数为 $\{N_i^h(\xi)\}_{i=1}^{n'}$ 和 $\{N_j^h(\eta)\}_{j=1}^{m'}$; ξ 和 η 方向上的基函数转换矩阵分别为 \hat{R} 和 \tilde{R} .由式(1)可得到二维 B 样条基函数的转换关系式:

$$N_i^H(\xi) N_j^H(\eta) = \left(\sum_s \hat{R}_{i,s} N_s^h(\xi) \right) \left(\sum_t \tilde{R}_{j,t} N_t^h(\eta) \right) = \sum_s \sum_t \hat{R}_{i,s} \tilde{R}_{j,t} N_s^h(\xi) N_t^h(\eta). \tag{3}$$

对疏密网格的自由度进行全局编号,则可得到总体自由度的转换矩阵的元素为

$$R_{\alpha,\beta} = \hat{R}_{i,s} \tilde{R}_{j,t}, \quad \alpha = i + (j - 1)n, \beta = s + (t - 1)n'. \tag{4}$$

疏密网格之间的传递通过网格转换矩阵实现:

$$\mathbf{u}^H = \mathbf{R}\mathbf{u}^h, \mathbf{u}^h = \mathbf{P}\mathbf{u}^H, \quad (5)$$

式中, \mathbf{u}^H 是疏网格上的未知量, \mathbf{u}^h 是密网格上的未知量.

1.2 多重网格算法

利用节点插入得到的疏密网格以及网格转换矩阵, 可以建立求解线性方程的多重网格迭代. 假设有两组网格, 疏网格到密网格的转换矩阵为 \mathbf{P} , 密网格到疏网格的转换矩阵为 \mathbf{R} , 在密网格上应用等几何方法得到如下代数方程组:

$$\mathbf{A}^h \mathbf{u}^h = \mathbf{f}^h, \quad (6)$$

式中, \mathbf{A}^h 是刚度阵, \mathbf{f}^h 是右端项. 本文考虑刚度阵为对称正定的情况.

该方程可用加权 Jacobi 迭代, Gauss-Seidel 迭代和 SOR 迭代等常用迭代方法求解, 这些迭代方法都属于 Richardson 型迭代, 可统一写成

$$\mathbf{u}_{m+1}^h = (\mathbf{I} - \mathbf{B}^h \mathbf{A}^h) \mathbf{u}_m^h + \mathbf{B}^h \mathbf{f}^h, \quad (7)$$

式中, \mathbf{u}_m^h 是第 m 次迭代的近似解, \mathbf{I} 是单位阵, \mathbf{B}^h 是预处理矩阵.

代数方程组的误差 $\mathbf{e}_m^h = \mathbf{u}^h - \mathbf{u}_m^h$ 和余量 $\mathbf{r}_m^h = \mathbf{B}^h - \mathbf{A}^h \mathbf{u}_m^h$ 之间满足如下余量方程:

$$\mathbf{A}^h \mathbf{e}_m^h = \mathbf{r}_m^h. \quad (8)$$

采用 Richardson 型迭代求解线性方程组时, 迭代过程中误差的高频分量衰减速度快, 而低频分量衰减速度慢. 随着迭代的进行, 误差中的低频分量将占主导地位, 并且难以衰减掉. 因为低频分量变化缓慢, 几何上较为光滑, 所以 Richardson 型迭代可看成一个误差光顺过程.

对高频误差分量, 需要用较密的网格以反映其分布; 但对光滑的低频误差, 可以用较疏的网格. 因此, 当 Richardson 型迭代收敛速度降低时, 可通过求解下面疏网格上的余量方程得到低频误差的值:

$$\mathbf{A}^H \mathbf{e}_m^H = \mathbf{r}_m^H, \quad (9)$$

式中, \mathbf{A}^H , \mathbf{e}_m^H , \mathbf{r}_m^H 分别为疏网格上的刚度阵、误差和余量. 疏网格上的误差和余量根据转换关系 (5) 得到; 刚度阵 \mathbf{A}^H 可通过在疏网格上用等几何方法得到, 也可以通过多重网格的 Galerkin 条件得到

$$\mathbf{A}^H = \mathbf{R}\mathbf{A}^h\mathbf{P}. \quad (10)$$

由于 \mathbf{A}^h 是对称正定的, 并且转换矩阵 \mathbf{R} 是行满秩的, 从式 (10) 得到的在疏网格上的刚度阵也是对称正定的. 求得方程 (9) 的解后, 将疏网格的误差转换到密网格上, 根据误差的定义来修正当前的近似解 \mathbf{u}_m^h . 修正之后, 一般再进行一次光顺操作进一步降低高频误差.

总体上, 多重网格迭代过程是在密网格上进行光顺, 去掉高频误差, 在粗网格上进行误差修正, 去掉低频误差. 如果疏网格的自由度已经很少, 那么可以直接求解疏网格上的余量方程; 如果疏网格的自由度仍然很大, 那么对余量方程 (9) 进行类似的光顺和修正过程, 直到网格足够稀疏可以直接求解.

算法 1 给出多重网格迭代过程的伪代码. 假设有 k 重网格, 第 $i+1$ 层网格是在第 i 层网格基础上通过节点插入得到, 该算法是从当前近似解 \mathbf{u}_k^m 得到下一个近似解 \mathbf{u}_k^{m+1} 的计算过程. 算法中 \mathbf{A}_k 为第 k 层网格下的刚度阵, \mathbf{R}_k^{k-1} 为第 k 层到第 $k-1$ 层的转换矩阵, \mathbf{P}_{k-1}^k 为第 $k-1$ 层到第 k 层的转换矩阵. \mathbf{S}^{μ_1} 表示进行 μ_1 次光顺, 光顺过程为 Richardson 型迭代 (如加权 Jacobi, Gauss-Seidel, SOR 迭代), μ_1, μ_2 一般取 1 或 2, 参数 γ 表示粗网格修正的迭代次数, γ 决定了访问各层网格的路径, $\gamma=1$ 为 V 型循环, $\gamma=2$ 为 W 型循环, 当采用 3 层网格时, 循环过程如图 1 所示.

算法 1 $\mathbf{u}_k^{m+1} = \text{MG}(\mathbf{A}_k, \mathbf{f}_k, \mathbf{u}_k^m, k, \gamma, \mu_1, \mu_2)$

IF $k = 1$

$$\mathbf{u}_k^{m+1} = \mathbf{A}_k^{-1} \mathbf{f}_k$$

ELSE

Presmoothing:

$$\mathbf{u}_k^{m+1} = \mathbf{S}^{\mu_1}(\mathbf{A}_k, \mathbf{f}_k, \mathbf{u}_k^m)$$

Correction:

$$\mathbf{d}_{k-1} = \mathbf{R}_k^{k-1}(\mathbf{f}_k - \mathbf{A}_k \mathbf{u}_k^m)$$

$$\mathbf{v}_{k-1} = \text{MG}^\gamma(\mathbf{R}_k^{k-1} \mathbf{A}_k \mathbf{P}_{k-1}^k, \mathbf{d}_{k-1}, \mathbf{0}, k-1, \gamma, \mu_1, \mu_2)$$

$$\mathbf{u}_k^{m+1} = \mathbf{u}_k^{m+1} + \mathbf{P}_{k-1}^k \mathbf{v}_{k-1}$$

Postsmoothing:

$$\mathbf{u}_k^{m+1} = \mathbf{S}^{\mu_2}(\mathbf{A}_k, \mathbf{f}_k, \mathbf{u}_k^{m+1})$$

END

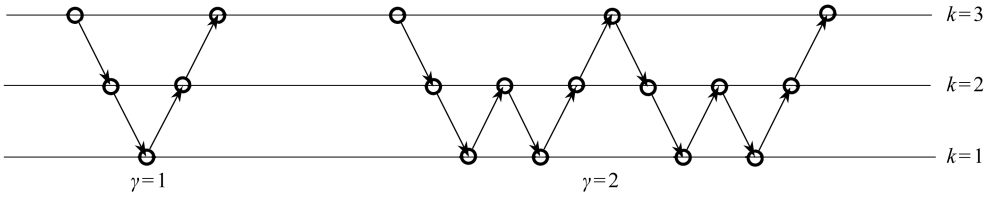


图1 多重网格循环形式

Fig.1 Multigrid cycles

2 多重网格共轭梯度法

多重网格迭代可以看成静态迭代方法,它采用的预处理矩阵具有特殊的形式,将该预处理矩阵用于共轭梯度法即形成了多重网格共轭梯度法。

采用算法 1 对方程 (6) 进行迭代求解时,误差向量的迭代关系为^[7]

$$\mathbf{e}_k^{m+1} = \mathbf{M}_s \mathbf{e}_k^m, \tag{11}$$

其中迭代矩阵 \mathbf{M}_s 由下面的递归关系给出:

$$\begin{cases} \mathbf{M}_1 = \mathbf{0}, \\ \mathbf{M}_k = \mathbf{S}^{\mu_2}(\mathbf{I}_k - \mathbf{P}_{k-1}^k (\mathbf{I}_{k-1} - \mathbf{M}_{k-1}^\gamma) \mathbf{A}_{k-1}^{-1} \mathbf{R}_k^{k-1} \mathbf{A}_k) \mathbf{S}^{\mu_1}. \end{cases} \tag{12}$$

如果把多重网格迭代看成静态迭代方法,那么它可以写成与式 (7) 类似的矩阵形式:

$$\mathbf{u}_k^{m+1} = (\mathbf{I} - \mathbf{B}_k \mathbf{A}_k) \mathbf{u}_k^m + \mathbf{B}_k \mathbf{f}_k. \tag{13}$$

由式 (13) 得到误差向量的迭代关系为

$$\mathbf{e}_k^{m+1} = (\mathbf{I} - \mathbf{B}_k \mathbf{A}_k) \mathbf{e}_k^m. \tag{14}$$

比较式 (11) 和 (14) 可以看出,多重网格迭代采用的预处理矩阵为

$$\mathbf{B}_k = (\mathbf{I}_k - \mathbf{M}_k) \mathbf{A}_k^{-1}. \tag{15}$$

虽然 \mathbf{B}_k 的表达式包含了 \mathbf{A}_k 的逆矩阵和复杂的递归矩阵 \mathbf{M}_k , 无法直接生成,但是它仍然可用作共轭梯度法的预处理矩阵.这是由于共轭梯度法只要求预处理矩阵对称正定,能快速计算出与任意向量的乘积,并不需要生成预处理矩阵.当 $\mu_1 = \mu_2$ 时,采用 Jacobi 光滑或对称 Gauss-Seidel 光滑的 V 型和 W 型多重网格算法都满足对称正定要求^[12].由式 (13) 可知 \mathbf{B}_k 与任意向量 \mathbf{v} 的乘积 $\mathbf{B}_k \mathbf{v}$ 可快速计算出,实际上只要以 $\mathbf{0}$ 向量为初始值,以 \mathbf{v} 为右端项应用算法 1 进行 1 次多重网格迭代即可.算法 2 是求解线性方程 (6) 的多重网格共轭梯度法伪代码,它与一般共轭梯度算法的不同是在预处理部分调用了算法 1 的多重网格迭代。

算法 2 MGCG($A_k, f_k, u, k, \gamma, \mu_1, \mu_2$)

$$r = f_k - A_k u$$

$$z = \text{MG}(A_k, r, \mathbf{0}, k, \gamma, \mu_1, \mu_2)$$

$$p = z$$

WHILE not converges

$$\tau = r^T z$$

$$w = A_k p$$

$$\alpha = \tau / (w^T p)$$

$$u = u + \alpha p$$

$$r = r - \alpha w$$

$$z = \text{MG}(A_k, r, \mathbf{0}, k, \gamma, \mu_1, \mu_2)$$

$$\beta = r^T z / \tau$$

$$p = z + \beta p$$

END

3 收敛性分析

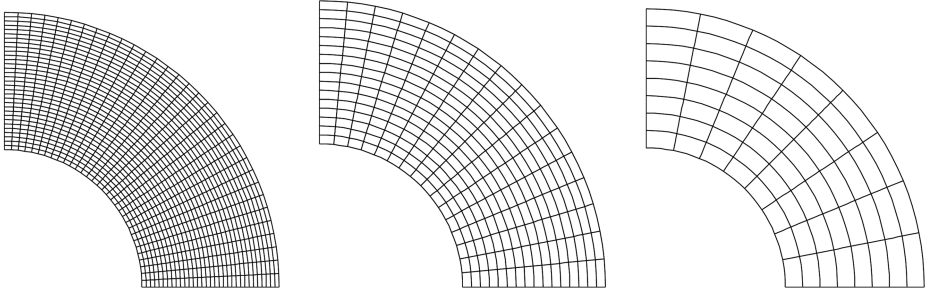
为验证迭代的收敛性, 本文对文献[14]的程序进行了扩展, 实现了多重网格算法和多重网格共轭梯度算法, 并用该程序求解了二维和三维区域上的 Poisson 方程。

Poisson 方程的形式为

$$\begin{cases} \Delta u(\mathbf{x}) = f, & \mathbf{x} \in \Omega, \\ u(\mathbf{x}) = g, & \mathbf{x} \in \Gamma_D, \\ \partial u(\mathbf{x}) / \partial \mathbf{n} = h, & \mathbf{x} \in \Gamma_N, \end{cases} \quad (16)$$

式中 u 是待求的未知函数, Ω 是二维或三维的计算区域, Γ_D 和 Γ_N 分别为 Dirichlet 边界和 Neumann 边界, $\Gamma_D \cup \Gamma_N = \partial\Omega$, \mathbf{n} 是边界的外法线方向, \mathbf{x} 是计算区域中的一点, f, g 和 h 是给定的函数。

本文用文献[10]给出的二维和三维算例作为研究对象进行分析. 对二维问题, 所考虑的计算区域为第 I 象限内的圆环, 内外径分别为 1 和 2, 取解析解为 $u = (x^2 + y^2 - 3\sqrt{x^2 + y^2} + 2)\sin(2\arctan(y/x))$; 对三维问题, 所考虑的计算区域为单位立方体, 取解析解为 $u = \sin(\pi x)\sin(\pi y)\sin(\pi z)$. 方程(16)的右端项及边界条件根据解析解给出。



(a) 32×32

(b) 16×16

(c) 8×8

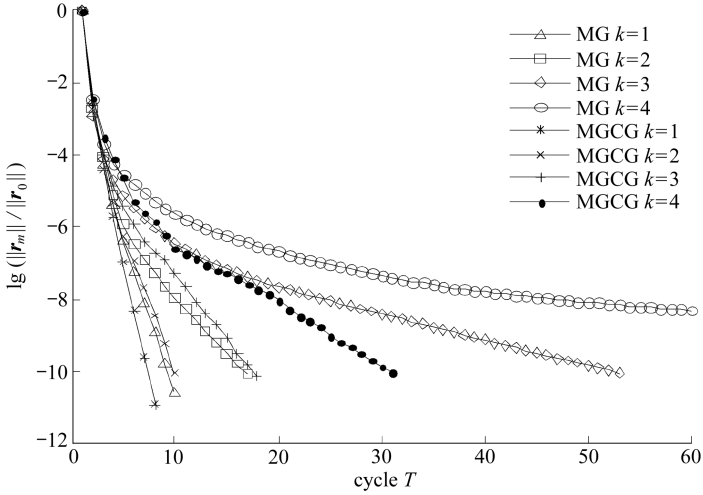
图2 二维问题的网格

Fig.2 Grids for the 2D problem

本文的所有算例都以 $\mathbf{0}$ 向量为初始值进行迭代求解,收敛条件统一取为

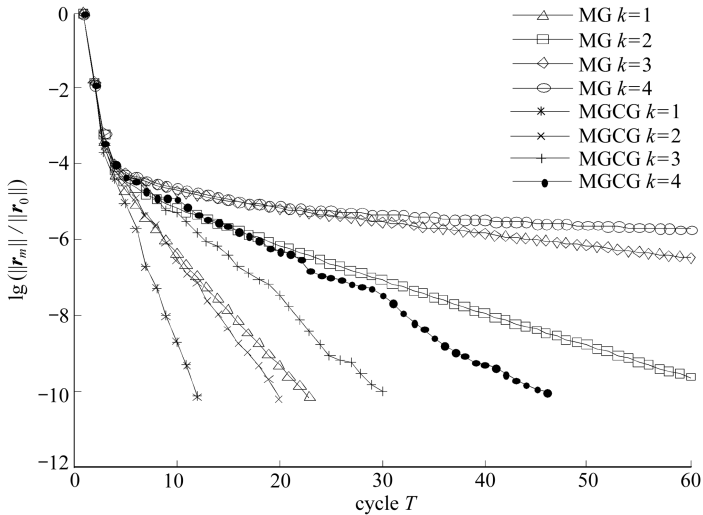
$$\|\mathbf{r}_m\| / \|\mathbf{r}_0\| \leq 10^{-10}. \tag{17}$$

对二维 Poisson 方程分别考虑 Dirichlet 边值问题和混合边值问题,前者所有边界施加 Dirichlet 边界条件,后者在圆环的外径处施加 Neumann 边界条件 $\partial u / \partial \mathbf{n} = (2\sqrt{x^2 + y^2} - 3)\sin(2\arctan(y/x))$, 其它边界施加 Dirichlet 边界条件.分别采用算法 1 (MG 迭代) 和算法 2 (MGCG 迭代) 求解代数方程, MG 和 MGCG 都采用了 3 层网格 (图 2), 多重网格循环型式为 $V(1,1)$, 即 $\gamma = \mu_1 = \mu_2 = 1$, 光顺算法为对称 Gauss-Seidel 迭代.



(a) Dirichlet 边值问题

(a) The Dirichlet boundary value problem



(b) 混合边值问题

(b) The mixed boundary value problem

图 3 二维问题 MG 和 MGCG 迭代收敛性对比

Fig.3 Convergence comparison between MG and MGCG for the 2D problems

图 3 是二维问题的迭代收敛过程,其中 k 是 NURBS 基函数的可导阶数.从图中可以看出,不论是 Dirichlet 边值问题还是混合边值问题,随着迭代的进行 MG 的收敛速度趋于恒定,在图

中为直线,但直线的斜率随 k 的增大而减小.MGCG 的收敛速度虽然不趋于恒定,但是在 k 相同时 MGCG 比 MG 的收敛速度快.当 $k = 1$ 时,考虑到 MGCG 每一次循环的计算量大于 MG,这两种算法在效率上相当;随着 k 的增加,MGCG 的收敛速度明显高于 MG.因此,如果通过增加 k 来提高数值解的精度,那么 MGCG 在效率上优于 MG.比较图 3(a) 和图 3(b) 可知当基函数的可导阶数 k 一定时,混合边值问题求解时 MGCG 迭代次数比 Dirichlet 边值问题的迭代次数多.虽然迭代次数有所增加,但方程仍然可以高效求解,表明 MGCG 迭代对不同的边界条件有良好的适应性.

表 1 给出了在式 (17) 的收敛条件下二维 Dirichlet 边值问题 MG 迭代和 MGCG 迭代所需的 CPU 时间,算法用 MATLAB 语言编程实现,计算采用 Intel Core2 CPU,处理器频率 2.33 GHz,表中所列时间是进行了 10 次计算的均值.从表中可以看出,随着 k 的增加,对二维和三维问题, MG 迭代和 MGCG 迭代的时间都会增加,这是由于 k 增大导致系数矩阵的条件数增大以及自由度增多引起的.当 k 相同时, MGCG 迭代比对应的 MG 迭代时间少,这种相对优势当 k 越大时越明显.

表 1 MG 迭代和 MGCG 迭代的 CPU 时间 (t/s)
Table 1 CPU time for MG and MGCG iterations (t/s)

k	2-dimensional		3-dimensional	
	MG	MGCG	MG	MGCG
1	0.055	0.049	0.946	0.799
2	0.170	0.108	13.300	5.478
3	0.861	0.300	96.646	27.021
4	4.244	0.780	-	-

对三维 Poisson 方程本文求解了 Dirichlet 边值问题,图 4 是 MG 和 MGCG 的迭代收敛过程,采用了 $V(2,2)$ 型多重网格循环,最密的网格为 $16 \times 16 \times 16$,共 3 层网格.从图中可以看出对三维问题 MGCG 也比 MG 的收敛性高.比较图 3 和图 4 可知,三维问题比二维问题更难收敛,对相同的 k 值三维问题比二维问题需要更多的迭代,因此 MGCG 对三维问题更具有意义.

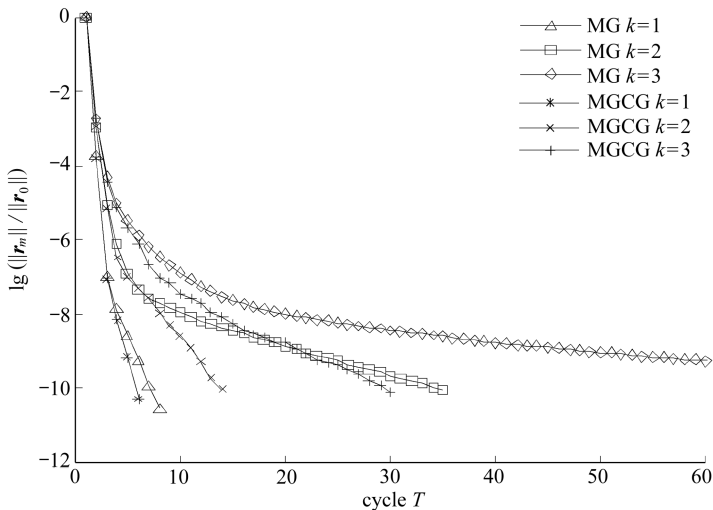


图 4 三维问题 MG 和 MGCG 的收敛性对比

Fig.4 Convergence comparison between MG and MGCG for the 3D problem

下面通过二维 Dirichlet 问题来研究多重网格参数和最密网格的密度对算法的影响.取 $k =$

4, 最密网格 32×32 , 采用 3 层网格, 图 5 中给出了多重网格循环形式对 MGCG 收敛性的影响。从图中可以看出增加光顺次数可以提高收敛的速度, 这是由于增加光顺次数提高了高频误差的衰减速度, 光顺次数一定时 V 型循环和 W 型对 MGCG 收敛性的影响很小。

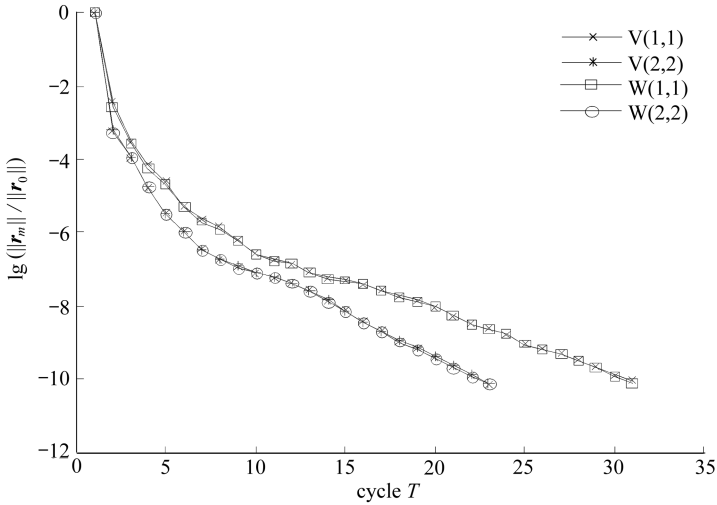


图 5 多重网格循环参数对 MGCG 收敛性的影响

Fig.5 Influence of multigrid cycle parameters on MGCG convergence

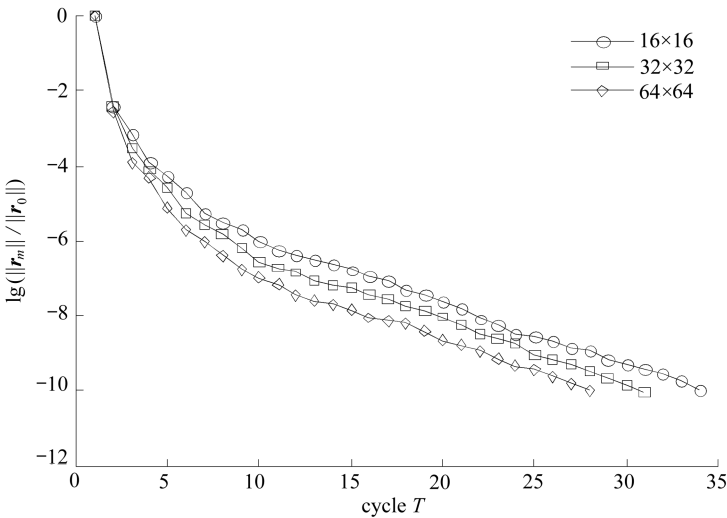


图 6 网格密度对 MGCG 收敛性的影响

Fig.6 Influence of grid density on MGCG convergence

图 6 是网格密度对 MGCG 收敛性的影响, 分别计算了最密网格为 16×16 , 32×32 , 64×64 的 3 种情况, 对应的多重网格层数为 2, 3, 4, NURBS 基函数的光滑性参数 $k = 4$ 。可以看出随着网格密度的增加, 总的迭代次数减少, 这主要是由于最初几次迭代中较密网格的误差下降更快, 随着迭代进行不同网格密度的收敛速度趋于一致。

以上算例表明, NURBS 基函数阶数对 MGCG 迭代收敛速度影响较大, 网格规模对 MGCG 迭代收敛速度影响较小。当网格数一定时, 随着 NURBS 基函数阶数增加所需的迭代次数增加明显(图 3、图 4); 当基函数阶数一定时, 随着网格规模增大所需的迭代次数基本不变(图 6)。

4 结 论

多重网格方法是一种求解代数方程的高效迭代方法,在等几何分析中也可以应用多重网格技术.等几何分析采用了基于 NURBS 的基函数,尺寸不同的疏密网格由节点插入算法生成,网格间的转换关系由离散 B 样条得到.但是随着 NURBS 基函数阶的增加, Jacobi 迭代或 Gauss-Seidel 迭代的光顺效果变差,进而多重网格迭代收敛速度降低.

本文建立了一种适用于等几何分析的多重网格共轭梯度法.该方法用共轭梯度法作为基础迭代算法,用多重网格方法进行预处理.多重网格方法可看成用简单迭代方法求解某个预处理矩阵作用下的代数方程,这个预处理矩阵满足共轭梯度法的要求.该方法结合了共轭梯度法与多重网格方法的优点,当 NURBS 基函数阶数增加时,在多重网格中衰减较慢的误差可在共轭梯度法中快速衰减.用它求解了二维和三维 Poisson 方程,结果表明它比多重网格法的收敛速度更快,当 NURBS 基函数阶数较高或者计算三维问题时本文方法优于多重网格方法.

参考文献 (References):

- [1] Hughes T J R, Cottrell J A, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement[J]. *Computer Methods in Applied Mechanics and Engineering*, 2005, **194**(39/41): 4135-4195.
- [2] 陈德祥, 徐自力, 刘石, 冯永新. 求解 Stokes 方程的最小二乘等几何分析方法[J]. 西安交通大学学报, 2013, **47**(5): 51-55. (CHEN De-xiang, XU Zi-li, LIU Shi, FENG Yong-xin. Least squares isogeometric analysis for Stokes equation[J]. *Journal of Xi'an Jiaotong University*, 2013, **47**(5): 51-55. (in Chinese))
- [3] Evans J A, Bazilevs Y, Babuska I, Hughes T J R. n -widths, sup-infs, and optimality ratios for the k -version of the isogeometric finite element method[J]. *Computer Methods in Applied Mechanics and Engineering*, 2009, **198**(21/26): 1726-1741.
- [4] Lipton S, Evans J A, Bazilevs Y, Elguedj T, Hughes T J R. Robustness of isogeometric structural discretizations under severe mesh distortion[J]. *Computer Methods in Applied Mechanics and Engineering*, 2010, **199**(5/8): 357-373.
- [5] 陶文铨. 计算传热学的近代进展[M]. 北京: 科学出版社, 2000. (TAO Wen-quan. *Advances in Computational Heat Transfer* [M]. Beijing: Science Press, 2000. (in Chinese))
- [6] Kumar D S, Kumar K S, Das M K. A fine grid solution for a lid-driven cavity flow using multi-grid method[J]. *Engineering Applications of Computational Fluid Mechanics*, 2009, **3**(3): 336-354.
- [7] Trottenberg U, Oosterlee C W, Schuller A. *Multigrid* [M]. Academic Press, 2000.
- [8] Briggs W L, Henson V E, McCormick S F. *A Multigrid Tutorial* [M]. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000.
- [9] Sampath R S, Biros G. A parallel geometric multigrid method for finite elements on octree meshes[J]. *SIAM Journal on Scientific Computing*, 2010, **32**(3): 1361-1392.
- [10] Gahalaut K P S, Kraus J K, Tomar S K. Multigrid methods for isogeometric discretization[J]. *Computer Methods in Applied Mechanics and Engineering*, 2013, **253**: 413-425.
- [11] Saad Y. *Iterative Methods for Sparse Linear Systems* [M]. SIAM, 2003.
- [12] Tatebe O. The multigrid preconditioned conjugate gradient method [C] // Melson N D. *The Sixth Copper Mountain Conference on Multigrid Methods*. Copper Mountain: NASA, 1993:

621-634.

- [13] Cohen E, Lyche T, Riesenfeld R. Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics[J]. *Computer Graphics and Image Processing*, 1980, **14**(2): 87-111.
- [14] de Falco C, Reali A, Vazquez R. GeoPDEs: a research tool for isogeometric analysis of PDEs [J]. *Advances in Engineering Software*, 2011, **42**(12): 1020-1034.

A Multigrid Preconditioned Conjugate Gradient Method for Isogeometric Analysis

LIU Shi¹, CHEN De-xiang², FENG Yong-xin¹, XU Zi-li², ZHENG Li-kun¹

(1. *Electric Power Research Institute, Guangdong Power Grid Corporation, Guangzhou 510080, P.R.China;*

2. *State Key Laboratory for Strength and Vibration of Mechanical Structures; School of Aerospace, Xi'an Jiaotong University, Xi'an 710049, P.R.China*)

Abstract: Accuracy of the isogeometric analysis can be improved through increase of the order of the NURBS basis function, but convergence of the multigrid will be slowed down at the same time. A method which combined the multigrid technique and preconditioned conjugate gradient iteration was proposed to accelerate the multigrid convergence. In the proposed method, the conjugate gradient part serves as the primary iteration, while the multigrid part serves as the preconditioner. The Poisson's equation was solved with the multigrid method and multigrid preconditioned conjugate gradient method respectively for comparison. The results show that the multigrid preconditioned conjugate gradient method converges faster than the multigrid method especially in the cases of high-order NURBS basis functions or 3-dimensional problems.

Key words: isogeometric analysis; multigrid; conjugate gradient; Poisson's equation; iterative algorithm; NURBS

Foundation item: The National Basic Research Program of China (973 Program) (2011CB706505); The National Natural Science Foundation of China (51275385)