求解二次 Lagrangian 有限元方程的代数两水平方法 *

李 明

(红河学院数学学院，蒙自 661199)

(E-mail: lm-001@126.com)

李郴良

(桂林电子科技大学数学与计算科学学院，桂林 541004)

崔向照

(红河学院数学学院，蒙自 661199)

摘 要 基于矩阵图集的粗化算法，构造一种新的插值算子，提出了瀑布型代数两重网格法；然后结合部分几何信息，提出了求解二次 Lagrangian 有限元方程的代数两水平方法. 数值实验表明该算法稳健性强、计算量更少.

关键词 二次 Lagrangian 有限元方程；插值算子；瀑布型代数两重网格法；代数两水平方法

MR(2000) 主题分类 65N30, 65N55

中图分类 O241

1 引言

代数多重网格 (AMG) 法 [1,2]是求解偏微分方程离散化方程的一类有效的方法. Lagrangian 有限元法是求解偏微分方程的一类重要的离散化方法，可分为线性 Lagrangian 有限元和高次 Lagrangian 有限元. 对于系数矩阵稠密性更强的高次有限元方程, 通常的 AMG 法很难控制粗网格的自由度和合理设计插值算子，因此借助部分几何或分析信息，再通过代数途径来构造相应的高效 AMG 法是目前 AMG 法发展的一个新趋势 [3−6]. 为了求解高次 Lagrangian 有限元方程，舒适教授等提出一类基于部分几何和分析信息的代数多重网格法 [4−6].

这类算法的主要思想如下：结合网格几何信息，通过代数途径建立高次 Lagrangian 有

本文 2011年 1 月 9 日收到. 2013 年 6 月 14 日收到修改稿.

* 国家自然科学基金 (11161014), 云南省科技厅青年 (2012FD054) 和红河学院硕博 (XJ1S0925) 资助项目.

限元空间 V_h^{sec} 到线性 Lagrangian 有限元空间 V_h^{lin} 的网格粗化算法和提升算子，使用通常的 AMG 法求解 V_h^{lin} 对应的粗网格方程，建立求解高次 Lagrangian 有限元方程的代数多重网格法.

为了减少求解 V_h^{lin} 对应的粗网格方程的计算量以实现整个问题的快速求解，本文基于 [7] 中的矩阵图集粗化算法，构造一种插值算子，提出一种瀑布型代数两重网格法，接着结合 [4–6] 中基于部分几何信息的两层网格法，提出了一种求解二次 Lagrangian 有限元方程的代数两水平方法. 数值实验表明该算法计算量更少，具有较强的稳健性. 本文的思想可以推广至其他离散格式.

2　模型问题及相关算法

考虑如下模型问题

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases} \tag{1}$$

其中 f 为已知函数，$u \in H_0^1(\Omega)$ 为待求函数，Ω 是一个凸多边形区域，$\partial\Omega$ 是 Ω 的边界.

采用均匀网格剖分，分别使用线性 Lagrangian 有限元、二次 Lagrangian 有限元离散问题 (1), 形成有限元空间 V_h^{lin}, V_h^{sec}, 对应的线性 Lagrangian 有限元方程，二次 Lagrangian 有限元方程分别为

$$A_h^{\text{lin}} U_h^{\text{lin}} = F_h^{\text{lin}}, \tag{2}$$

$$A_h^{\text{sec}} U_h^{\text{sec}} = F_h^{\text{sec}}. \tag{3}$$

对于线性 Lagrangian 有限元方程 (即方程 (2)), 其刚度矩阵对应的网格图和相应的几何网格图基本一致，使用通常的 AMG 法求解方程 (2) 具有计算量少、稳健性强的优点，该类算法具体过程如下：

算法 1　V 型代数多重网格法 [4].

for　$l = 1, \cdots, J-1$

　　$U_l := U_l + G_l(F_l - A_l U_l), \quad j = 1, \cdots, m_1;$

　　$F_{l+1} := (P_{l+1}^l)^T (F_l - A_l U_l);$

end

$U_J := (A_J)^{-1} F_J;$

for　$l = J-1, \cdots, 1$

　　$U_l := U_l + P_{l+1}^l U_{l+1};$

　　$U_l := U_l + G_l(F_l - A_l U_l), \quad j = 1, \cdots, m_2;$

end

其中，$F_1 := F_h^{\text{lin}}, A_1 := A_h^{\text{lin}}$, P_{l+1}^l 是通过代数途径建立起的插值算子，m_1, m_2 分别是前后光滑步数，G_l 为某种迭代算子.

对于二次 Lagrangian 有限元方程 (即方程 (3)), 其刚度矩阵的稠密性更强，很难控制粗网格的自由度和合理设计提升算子. 因此算法 1 对于方程 (3) 的应用效果差 (详见 [4,5]). 为此，[4] 结合部分几何信息，通过代数途径建立二次有限元空间 V_h^{sec} (细空间) 到线性有

限元空间 V_h^{lin}(粗空间) 的粗化算法和提升算子 P_q^l, 形成求解方程 (3) 的两层网格法. 具体如下

算法 2 两层网格法 [4].

步骤 1 (预处理).

(1) 构造限制算子 P_q^l (详见 [4]);

(2) 求提升算子 $P_l^q := {P_q^l}^T$;

(3) 求粗化矩阵 $A_H := P_q^l A_h^{\sec} P_l^q$;

步骤 2 前光滑 $U_h := U_h + G(F_h^{\sec} - A_h^{\sec} U_h^{\sec}), \quad j = 1, \cdots, m_3$;

步骤 3 精确求解粗网格方程: $A_H e_H = P_q^l (F_h^{\sec} - A_h^{\sec} U_h^{\sec})$ 得 e_H;

步骤 4 校正: $U_h^{\sec} := U_h^{\sec} + P_l^q e_H$;

步骤 5 后光滑 $U_h^{\sec} := U_h^{\sec} + G(F_h^{\sec} - A_h^{\sec} U_h^{\sec}), \quad j = 1, \cdots, m_4$;

其中, G 为迭代算子.

为了降低步骤 3 中精确求解粗网格方程的计算量, [4,5] 提出调用 k 次算法 1 来近似求解 $A_H e_H = P_q^l (F_h^{\sec} - A_h^{\sec} U_h^{\sec})$, 进而构造了一种求解方程 (3) 的代数多重网格法, 具体如下:

算法 3 代数多重网格法 [4,5].

步骤 1 (预处理).

(1) 构造限制矩阵 P_q^l(详见文献 [4]) ;

(2) 求提升算子 $P_l^q := {P_q^l}^T$;

(3) 求粗化矩阵 $A_H := P_q^l A_h^{\sec} P_l^q$;

步骤 2. 前光滑 $U_h := U_h + G(F_h^{\sec} - A_h^{\sec} U_h^{\sec}), \quad j = 1, \cdots, m_3$;

步骤 3. 调用 k 次算法 1 近似求解粗网格方程: $A_H e_H = P_q^l (F_h^{\sec} - A_h^{\sec} U_h^{\sec})$ 得 e_H^*;

步骤 4. 校正: $U_h^{\sec} := U_h^{\sec} + P_l^q e_H^*$;

步骤 5. 后光滑 $U_h^{\sec} := U_h^{\sec} + G(F_h^{\sec} - A_h^{\sec} U_h^{\sec}), \quad j = 1, \cdots, m_4$.

由算法 3 的步骤 3 可以看出, 为了得到满足精度要求的数值解 e_H^*, 往往需要调用 k 次算法 1, 结合算法 1 的计算流程可知在第 1 层到第 $J-1$ 层上的光滑迭代数均为 $k(m_1+m_2)$ 次. 对于未知点最多的第 1 层, 光滑 $k(m_1+m_2)$ 次的计算量较大. 因此, 需要发展计算量更少的算法.

3 求解二次 Lagrangian 有限元方程的代数两水平法

[7] 指出, 基于矩阵图集的粗化算法的计算效率优于基于强弱连接关系的粗化算法. 因此, 本文选用 [7] 中的粗化算法形成粗点集. 先引入如下定义:

定义 1[7] 设 A 为 n 阶方阵, $J = [1, \cdots, n]$, 定义关于矩阵 $A = [a_{\alpha,\beta}]_{\alpha,\beta \in J}$ 的图集 $G(A)$:

$$G(A) = \big\{(\alpha, \beta) \in J \times J : a_{\alpha,\beta} \neq 0 \text{ 且 } \alpha \neq \beta\big\}, \tag{4}$$

其中, $\alpha \in J$ 称为顶点, $(\alpha, \beta) \in G(A)$ 称为以 α 起点, 以 β 为终点的边.

在定义 1 的基础上, 引入如下粗化算法:

算法 4[7] 基于矩阵图集的粗化算法.

步骤 1. 输入图集 $G(A)$, 并建立空粗点集 $C=[\ ]$;

步骤 2. 从 $G(A)$ 中选取从未被访问过的顶点 α:

(1) 将 α 加入到粗点集 C 中，在图集 $G(A)$ 中将 α 点标注 “被访问过”;

(2) 选取满足 $(\alpha,\beta)\in G(A)$ 的点 β, 将 β 点都标注为 “被访问过”;

步骤 3. 重复步骤 2, 直至 $G(A)$ 中所有点都被访问过.

通过算法 4, 可以得到图集 $G(A)$ 中的粗点集 C, 记粗点集 C 的余集为 $\widetilde{F}$, 即有 $C\cup\widetilde{F}=J$. 并将 C 和 $\widetilde{F}$ 中元素按升序排列.

记 $N_i=\{j:|a_{ij}|\neq 0,\ i\neq j\}$ 为 i 点的邻点集. 记 $N_i^C=C\cap N_i$ 为 i 点的所有邻点中粗点构成的集合. 记 $N_i^{\widetilde{F}}=\widetilde{F}\cap N_i$ 为 i 点所有邻点中细点构成的集合. $i\in C$ 在粗点集 C 中的序号记为 $s=\mathrm{ind}\,(i)$. 记粗点集 C 中元素个数为 N.

基于算法 4, 我们讨论插值算子的构造. 对于方程 (2), 记 $A_h^{\mathrm{lin}}:=(a_{ij})_{n\times n}$, $F_h^{\mathrm{lin}}:=[f_1,f_2,\cdots,f_n]^T$, $U_h^{\mathrm{lin}}:=[u_{h,1},u_{h,2},\cdots,u_{h,n}]^T$. 可得

$$\sum_{j=1}^{n}a_{ij}u_{h,j}=f_i,\qquad i=1,2,\cdots,n,\tag{5}$$

即

$$a_{ii}u_{h,i}+\sum_{j\in N_i}a_{ij}u_{h,j}=f_i,\qquad i=1,2,\cdots,n.\tag{6}$$

设使用 Galerkin 方法由方程 (2) 构造的粗网格方程对应的解 $u_H=[u_{H,1},u_{H,2},\cdots,u_{H,N}]^T$ 已经求出，则将其作为细网格上与之对应点的近似值. 即将 u_H 作为 U_h^{lin} 中粗点集 C 对应的解

$$u_{h,i}:\approx u_{H,s},\qquad s=\mathrm{ind}(i),\quad i\in C,\tag{7}$$

则对于 $\forall i\in\widetilde{F}$, 有

$$a_{ii}u_{h,i}+\sum_{j\in N_i^c}a_{ij}u_{h,j}+\sum_{t\in N_i^{\widetilde{F}}}a_{it}u_{h,t}=f_i,\tag{8}$$

可得

$$a_{ii}u_{h,i}+\sum_{t\in N_i^{\widetilde{F}}}a_{it}u_{h,t}=f_i-\sum_{j\in N_i^c}a_{ij}u_{H,\mathrm{ind}\,(j)},\qquad\forall i\in\widetilde{F}.\tag{9}$$

因为 $u\in H_0^1(\Omega)$, 根据连续性，当步长 h 足够小时，u_h 可做如下近似

$$u_{h,t}:\approx u_{h,i},\qquad\forall t\in N_i^{\widetilde{F}},\tag{10}$$

则 (9) 式可变为

$$\Big(a_{ii}+\sum_{t\in N_i^{\widetilde{F}}}a_{it}\Big)u_{h,i}\approx f_i-\sum_{j\in N_i^c}a_{ij}u_{H,\mathrm{ind}(j)},\qquad\forall i\in\widetilde{F}.\tag{11}$$

因为 $A_h^{\rm lin}$ 是对称正定的，所以 $(a_{ii}+\sum\limits_{t\in \widetilde{N_i^F}} a_{it})\neq 0$, 故

$$u_{h,i}\approx \frac{1}{a_{ii}+\sum\limits_{t\in \widetilde{N_i^F}} a_{it}}\Big(-\sum_{j\in N_i^c} a_{ij}u_{H,{\rm ind}\,(j)}+f_i\Big),\qquad \forall\, i\in \widetilde{F}. \tag{12}$$

将 (7) 式和 (12) 式综合起来

$$u_{h,i}:\approx\begin{cases} u_{H,s}, & \text{if}\ \ i\in C, s={\rm ind}(i),\\ \dfrac{1}{a_{ii}+\sum\limits_{t\in \widetilde{N_i^F}} a_{it}}\Big(-\sum\limits_{j\in N_i^c} a_{ij}u_{H,{\rm ind}(j)}+f_i\Big), & \text{if}\ \ i\in \widetilde{F}.\end{cases} \tag{13}$$

使用上述插值由 u_H 和 $F_h^{\rm lin}$ 得到细层初始值 U_h^0 的过程可记为

$$U_h^0=P_H^h u_H+\varpi_h(A_h^{\rm lin},F_h^{\rm lin}), \tag{14}$$

其中

$$P_H^h=(p_{ij})_{n\times N}=\begin{cases} 1, & \text{if}\ \ i\in C, j={\rm ind}(i),\\ \dfrac{-a_{ik}}{a_{ii}+\sum\limits_{t\in \widetilde{N_i^F}} a_{it}}, & \text{if}\ \ i\in \widetilde{F},\ k\in N_i^C,\ j={\rm ind}(k),\\ 0, & \text{other}\end{cases} \tag{15}$$

$$\varpi_h(A_h^{\rm lin},F_h^{\rm lin})=(\varpi_h)_{n\times 1}=\begin{cases} 0, & \text{if}\ \ i\in C,\\ \dfrac{f_i}{a_{ii}+\sum\limits_{t\in \widetilde{N_i^F}} a_{it}}, & \text{if}\ \ i\in \widetilde{F}.\end{cases} \tag{16}$$

为便于描述，记算法 2 的步骤 3 中粗网格方程 $A_He_H=P_q^l(F_h^{\rm sec}-A_h^{\rm sec}U_h^{\rm sec})$ 为

$$A_He_H=\Phi_H. \tag{17}$$

针对方程 (17), 结合上述插值过程，我们提出如下算法：

算法 5 瀑布型代数两重网格法.

步骤 1. (预处理): $A_M:=(P_H^h)^TA_HP_H^h$, $\Phi_M:=(P_H^h)^T\Phi_H$;

步骤 2. 精确求解方程 $A_Me_M=\Phi_M$ 得 e_M^*;

步骤 3. 插值： $e_H^0=P_H^he_M^*+\varpi_h(A_H,\Phi_H)$;

步骤 4. 如果 $\frac{||\Phi_H-A_He_H^0||}{||\Phi_H||}<\varepsilon_1$, 令 $e_H^*:=e_H^0$, 输出 e_H^*, 停机，否则转步骤 5;

步骤 5. 磨光 m 次 $e_H^*:=G^m(e_H^0)$, 直至 $\frac{||\Phi_H-A_He_H^*||}{||\Phi_H||}<\varepsilon_1$, 输出 e_H^*, 停机.

其中 ε_1 为精度控制条件，文中选取 $\varepsilon_1=10^{-6}$.

由上可以看出，算法 5 不需要限制和前光滑，因此其算法结构比算法 1 更简单.

基于 [4] 的思想，我们将算法 5 与算法 2 相结合，提出求解二次 Lagrangian 有限元方程的代数两水平方法. 算法过程如下：

算法 6 代数两水平方法.

步骤 1. (预处理)

(1) 构造限制算子 P_q^l (详见 [4]);

(2) 求提升算子 $P_l^q := P_q^{l^T}$;

(3) 求粗化矩阵 $A_H := P_q^l A_h^{\mathrm{sec}} P_l^q$;

步骤 2. 前光滑 $U_h := U_h + G(F_h^{\mathrm{sec}} - A_h^{\mathrm{sec}} U_h^{\mathrm{sec}}), \quad j = 1, \cdots, m_3$;

步骤 3. 用算法 5 近似求解粗网格方程： $A_H e_H = P_q^l (F_h^{\mathrm{sec}} - A_h^{\mathrm{sec}} U_h^{\mathrm{sec}})$ 得 e_H^*；

步骤 4. 校正： $U_h^{\mathrm{sec}} := U_h^{\mathrm{sec}} + P_l^q e_H^*$;

步骤 5. 后光滑 $U_h^{\mathrm{sec}} := U_h^{\mathrm{sec}} + G(F_h^{\mathrm{sec}} - A_h^{\mathrm{sec}} U_h^{\mathrm{sec}}), \quad j = 1, \cdots, m_4$.

可以看出，算法 6 与算法 3 的不同之处是步骤 3 中使用不同的代数多重网格法.

4 数值实验

为了验证本文算法的有效性，给出如下数值算例.

算例 1 $f = \pi^2 \sin(\pi x)(\mathrm{e}^y - 1)(1 - y^2) - \sin(\pi x)\mathrm{e}^y(1 - y^2) + 4\sin(\pi x)\mathrm{e}^y y + 2\sin(\pi x)(\mathrm{e}^y - 1)$, 区域 $\Omega : (0,1) \times (0,1)$, 真解 $u = \sin(\pi x)(\mathrm{e}^y - 1)(1 - y^2)$.

算例 2 $f = \frac{1}{(1+x)^2}(\mathrm{e}^{(\cos(\pi x)+1)} - 1)\sin^2(\pi y) + \frac{2}{(1+x)}\sin(\pi x)\pi \mathrm{e}^{(\cos(\pi x)+1)}\sin^2(\pi y) + \ln(1+x)\cos(\pi x)\pi^2 \mathrm{e}^{(\cos(\pi x)+1)}\sin^2(\pi y) - \ln(1+x)\sin^2(\pi x)\pi^2 \mathrm{e}^{(\cos(\pi x)+1)}\sin^2(\pi y) - 2\ln(1+x)(\mathrm{e}^{(\cos(\pi x)+1)} - 1)\cos^2(\pi y)\pi^2 + 2\ln(1+x)(\mathrm{e}^{(\cos(\pi x)+1)} - 1)\sin^2(\pi y)\pi^2$, 区域 $\Omega : (0,1) \times (0,1)$, 真解 $u = \ln(1+x)(e^{(\cos(\pi x)+1)} - 1)\sin^2(\pi y)$.

算例 3[8] $f = 2\pi^2 \sin(\pi x)\sin(\pi y)$, 区域 $\Omega : (0,1) \times (0,1)$, 真解 $u = \sin(\pi x)\sin(\pi y)$.

表 1 算例 1 的数值结果

	算法 3				算法 6			
Mesh	S	$\|\|U_h^{\mathrm{sec}} - u\|\|_a$	$\sum_{i=1}^{S} 6k_i$	time(s)	S	$\|\|U_h^{\mathrm{sec}} - u\|\|_a$	$\sum_{i=1}^{S} \bar{m}_i$	time(s)
32*32	6	2.08310e-003	138	2.437	6	2.08310e-003	0	1.241
64*64	6	5.20995e-004	138	11.74	6	5.20995e-004	0	8.104
128*128	5	4.25380e-005	138	130.1	5	1.30262e-004	0	93.29
256*256	5	3.25665e-005	120	1829	5	3.25665e-005	0	954.1

采用一致三角形网格剖分，使用线性 Lagrangian 三角元，二次 Lagrangian 三角元离散以上算例，文中磨光算子 G 选用高斯赛德迭代法. 算法 3 步骤 3 中使用的算法 1 的层数 J 取 2. 算法 1 的粗化算法选用算法 4, 插值算子选用本文提出的插值矩阵 (即式 (15)), 前后磨光步 m_1, m_2 均取 3. 算法 3 与算法 6 的外迭代终止条件为 $\frac{\|F_h^{\mathrm{sec}} - A_h^{\mathrm{sec}} U_h^{\mathrm{sec}}\|_2}{\|F_h^{\mathrm{sec}}\|_2} \le 10^{-8}$. m_3, m_4 均取 3. 为了便于描述，约定 “S” 表示调用算法 3 (或算法 6) 的次数； “Mesh” 表示二次 Lagrangian 三角元网格规模； “$\|U_h^{\mathrm{sec}} - u\|_a$” 表示最细网格层上算法求出的数值解与问题真解的能量范数误差； $\sum_{i=1}^{S} 6k_i$ 表示调用 S 次算法 3 时步骤 3 中使用算法 1 的总磨光步数； $\sum_{i=1}^{S} \overline{m}_i$ 表示调用 S 次算法 6 时步骤 3 中使用算法 5 的总磨光步数； “time(s)” 表示

计算时间 (秒).

由表 1– 表 3 可以看出，在同等计算规模下，算法 3 与本文提出的算法 6 具有相同的调用次数 S 和计算精度，两者均具有较强的稳健性，但算法 6 具有整体计算量更少 (步骤 3 中无需磨光步 $\sum\limits_{i=1}^{S} m_i = 0$), 计算时间更短的优点．此外，算法 3 和算法 6 均使用了本文提出的插值矩阵 P_H^h, 数值结果也验证本文提出的插值算子的有效性.

表 2 算例 2 的数值结果

	算法 3				算法 6			
Mesh	S	$\|\|U_h^{\mathrm{sec}} - u\|\|_a$	$\sum\limits_{i=1}^{S} 6k_i$	time(s)	S	$\|\|U_h^{\mathrm{sec}} - u\|\|_a$	$\sum\limits_{i=1}^{S} \overline{m}_i$	time(s)
32*32	6	1.43269e-002	144	1.490	6	1.43269e-002	0	1.287
64*64	5	3.59190e-003	114	9.843	5	3.59190e-003	0	8.577
128*128	5	8.98611e-004	114	117.3	5	8.98611e-004	0	107.9
256*256	5	2.24693e-004	114	1682	5	2.24693e-004	0	963.8

表 3 算例 3 的数值结果

	算法 3				算法 6			
Mesh	S	$\|\|U_h^{\mathrm{sec}} - u\|\|_a$	$\sum\limits_{i=1}^{S} 6k_i$	time(s)	S	$\|\|U_h^{\mathrm{sec}} - u\|\|_a$	$\sum\limits_{i=1}^{S} \bar{m}_i$	time(s)
32*32	6	6.05365e-003	132	1.312	6	6.05365e-003	0	1.220
64*64	6	1.51473e-003	132	9.922	6	1.51473e-003	0	8.838
128*128	6	3.78766e-004	132	122.6	6	3.78766e-004	0	109.3
256*256	6	9.46966e-005	132	1638	6	9.46966e-005	0	971.4

综上所述，本文设计的代数两水平方法 (算法 6) 可有效求解二次 Lagrangian 有限元方程.

参 考 文 献

[1] Brandt A, MeCormick S F, Ruge J W. Algebraic Multigrid for Differential Eigenproblems. *SIAM J. Sci. Stat. Comput.*, 1983, 4: 244–260

[2] 舒适，黄云清，阳莺．一类三维等代数结构面剖分下的代数多重网格算法． *计算物理*, 2005, 22(6): 488–492 (Shu S, Huang Y Q, Yang Y, et al. A Class of Algebraic Multigrid Algorithms with Three Dimensional Equal Algebraic Structures. *Chi J. Comp. Phys.*, 2005, 22(6): 488–492)

[3] Brandt M, Cleary A J, Falgout R D, et al. Algebraic Multigrid Based on Element Interpolation (AMGe). *SIAM J. Sci. Comput.*, 2000, 22(5): 1570–1592

[4] 舒适．几类基于几何和分析信息的代数多重网格法及其应用．湘潭：湘潭大学，博士论文． 2005 (Shu S. Geometry-analysis Based Algebraic Multigrid Method and Applications. Xiangtan: Xiangtan Univ., Doctoral Dissertation, 2005)

[5] 孙杜杜，舒适．求解三维高次拉格朗日有限元方程的代数多重网格法． *计算数学*, 2005, 27(1): 101–112

(Sun D D, Shu S. An Algebraic Mulgtigrid of the High Order Lagrangian Finite Element Equation in R^3. *Math. Nume.*, 2005, 27(1): 101–112)

[6] 刘轩，舒适. 非结构四边形二次 Lagrangian 有限元方程的代数多重网格法及收敛性分析. 高等学校计算数学学报, 2005, 27(4): 215–222
(Liu X, Shu S. An Algebraic Multigrid Method of Quadratic Lagrangian Finite Element Equations under the Unstructured Quadrilateral Grids and Analysis for Convergence. *Nume. Math. J. Chi. Univ.*, 2005, 27(4): 215–222)

[7] Kickinger F. Algebraic Multigrid for Discrete Elliptic Second-order Problems. Multigrid Methods V. Proceedings of the 5th European Multigrid Conference, Hackbusch W(ed.), Springer Lecture Notes in Computational Science and Engineering, Vol. 3., Berlin: Springer-Verlag, 1998, 157–172

[8] 谭敏，肖映雄，舒适. 一种各向异性四边形网格下的代数多重网格法. 湘潭大学自然科学学报, 2005, 27(1): 78–84
(Tan M, Xiao Y X, Shu S. An Algebraic Multigrid Method on Anisotropic Quadrilateral Grid. *Natu. Sci. J. Xiangtan Univ.*, 2005, 27(1): 78–84)

Algebraic Two Level Method for a Quadratic Lagrangian Finite Element Equation

LI Ming

(*Department of Mathematics, Honghe University, Mengzi* 661199)

(*E-mail: lm-001@126.com*)

LI Chenliang

(*School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin* 541004)

CUI Xiangzhao

(*Department of Mathematics, Honghe University, Mengzi* 661199)

Abstract A new interpolation operator is proposed by using the coarsening method only based on the graph of the matrix. Then a new cascadic algebraic two grid method is presented by using the new interpolation operator. And a new algebraic two level method for the quadratic Lagrangian finite element equation is given by combining the cascadic algebraic two grid method and some geometric information of the grids. The numerical experiment results show that the new methods are more efficient and robust.

Key words quadratic Lagrangian finite element equation; interpolation operator; cascadic algebraic two grid method; algebraic two level method

MR(2000) Subject Classification 65N22; 65N30; 65N55

Chinese Library Classification: O241.82