平面 *n* 边域上高品质四边网格生成方法

简 群[1)], 蔺宏伟[1,2)*], 曹 琦[1)], 卢兴江[1)]

[1)] (浙江大学数学科学学院 杭州 310027)
[2)] (浙江大学 CAD&CG 国家重点实验室 杭州 310058)
(hwlin@zju.edu.cn)

摘 要: 在有限元分析中, 四边网格比三角网格更难以生成, 特别是在具有复杂形状和拓扑结构的平面域上. 为此, 基于几何迭代算法, 提出一种在形状复杂和高亏格的 *n* 边平面域上生成高质量四边网格的方法, 并保证生成的四边网格不自交. 该方法以自适应像素化离散技术生成的四边网格作为初始网格, 网格边界迭代拟合至给定的平面区域边界, 其中每次边界迭代后, 通过分层的 Laplace 算子改变内部顶点的位置; 在迭代过程中, 网格顶点的移动都受到限制, 保证生成的网格严格不自交. 最后通过实验验证了文中算法的效率和有效性.

关键词: 四边网格生成; 不自交; 几何迭代; 有限元分析
中图法分类号: TP391.41

Generating Quality Guaranteed Quadrilateral Mesh on an *n*-sided Region

Jian Qun[1)], Lin Hongwei[1,2)*], Cao Qi[1)], and Lu Xingjiang[1)]

[1)] (*School of Mathematical Science, Zhejiang University, Hangzhou* 310027)
[2)] (*State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou* 310058)

Abstract: In finite element analysis, the generation of quadrilateral (quad) meshes is harder than that of triangular meshes, especially on planar regions with complicated shape and topology structure. In this paper, we developed an iterative method to produce quad meshes on *n*-sided connected planar regions with complicated geometric shape and high genus, and the generated quad mesh is guaranteed to be non-self-overlapping. Starting with an initial quad mesh, which is constructed by adaptive pixelization, the boundary of the quad mesh is iteratively fitted to the boundary of the given planar region. After each iteration of the boundary, the positions of inner vertices are changed by the layered Laplace operation. Finally, the quad mesh is generated by further optimizations. In the iterations, the movements of the mesh vertices are restricted so that the produced quad mesh is guaranteed to be strictly non-self-overlapping. Lots of examples presented in this paper show the efficiency and effectiveness of the developed method.

Key words: quad mesh generation; non-self-overlapping; geometric iteration; finite element analysis

在有限元分析中, 对于相同的网格顶点, 四边网格的计算误差要小于三角形网格. 因此, 四边网格比三角形网格更理想. 但是, 四边网格的生成却比三角形网格难得多, 特别是对于 n 边平面区域 ($n\neq 4$).

目前, 虽然在有限元和相关领域中已有很多生成四边网格的方法, 但只有很少一部分能在 n 边平面域上生成四边网格($n\neq 4$). Gregory 曲面片映射方法(gregory patch mapping, GPM)[1], 首先构造

收稿日期: 2016-04-15; 修回日期: 2016-08-12. 基金项目: 国家自然科学基金(61272300, 61379072). 简 群(1991—), 女, 硕士, 主要研究方向为计算机图形学; 蔺宏伟(1973—), 男, 博士, 副研究员, 博士生导师, CCF 会员, 论文通讯作者, 主要研究方向为几何设计; 曹 琦(1992—), 男, 硕士研究生, 主要研究方向为几何迭代; 卢兴江(1964—), 男, 博士, 教授, 主要研究方向为几何计算.

一个Gregory曲面片去拟合一个n边平面区域($n>4$)，然后将 Gregory 曲面片的参数域上的参数网格映射到n边区域，形成一个结构网格．然而，GPM方法只能处理亏格为0的n边平面区域；甚至，由于生成的结构网格的不自交性质是由数值优化过程保证的，所以当数值优化过程在某些特殊情况下失效时，该方法在这些特殊情况下不能生成不自交网格[1]．另一方面，间接方法通常是用前沿技术[2-3]把一个三角网格转换成四边网格，然而这些方法不仅容易遗留孤立的三角形，而且生成的四边网格的质量严重依赖于原始三角网格．还有一些其他方法首先把n边区域($n>4$)分割成一些小四边区域，然后在每一小四边域上构造结构网格[4-5]．但是，n边区域($n>4$)的分割也不是一件很容易的事．

基于几何迭代算法，本文提出了一种在n边区域中生成有质量保证的四边网格的方法．给定一个 n 边平面区域，首先采用像素化技术将区域离散，生成一个初始的四边网格．在进行自适应调整之后，初始四边网格的边界迭代拟合至n边区域的边界．在每步迭代后，通过 Laplace 光滑算子将四边网格边界顶点的移动扩散到网格内部顶点．在以上 2 个步骤中，顶点的移动限制在可行域中，以保证网格的不自交性质．当网格边界和n边区域边界之间的误差达到给定精度时，迭代结束．本文方法能处理带有复杂形状和复杂拓扑结构(高亏格)的 n 边平面区域；生成的四边网格的质量有保证的，如严格不自交；在大多数顶点中，网格是正交的或接近正交的．

1 相关工作

如上所述，大多数四边网格生成方法都聚焦于四边域(n=4)上的网格生成．这些方法大致可以分为 2 类：一类是基于偏微分方程的方法(partial differential equation, PDE)[6-14]；另一类是基于边界保持映射(boundary-con-forming mapping, BCM)的方法[15-21]．

基于 PDE 的方法首先在区域边界分布一些离散点，然后通过求解一个椭圆型方程生成平面四边域上的四边网格[6-7,9-10]．这类方法生成网格的正则性质和光滑性质受数值方法和控制函数的限制[8,11-14]．一般来说，基于PDE的方法不能保证网格的不自交性质．此外，由于只能用数值方法求解椭圆PDE，且只能得到四边域离散点上的解，会导致基于 PDE 的方法运行较慢，且只能产生离散的四边网格．有关更多基于 PDE 的网格生成方法，见文献[22-23]．

另一方面，基于 BCM 的方法首先构造一个从矩形参数域到四边平面域的连续映射，通过把矩形参数域上的参数网格映射到四边域上，形成四边网格[15-16]．虽然基于 BCM 的方法一般都具有很好的鲁棒性，且易于实现，但它们很难确保生成的四边网格严格不自交[20-21]．这是因为确保不自交性质等价于构造一个连续双射，验证连续映射的双射性质涉及到一个复杂的高度非线性问题．另外，基于 BCM 的方法一般采用数值方法生成不自交的四边网格，但数值方法在某些情况下会失效．

上面介绍的方法都是在四边平面域上生成四边网格，而在 $n(n>4)$边平面域上生成四边网格比较困难．通常的方法需要首先将 $n(n>4)$边平面域分割成多个四边平面域[4-5]，然后在每个四边域上构造四边网格．但是区域分割也是一个困难的问题．也有些方法把三角网格转化成四边网格[2-3]．另外，利用改进的铺设算法可以生成分布均匀的四边网格[24]．进一步，基于Q-Morph算法，通过添加约束条件可以提高网格的均匀性[25]．最后，虽然 GPM 方法[1]能直接在 $n(n>4)$边平面域上生成四边网格，但它不能处理有洞的区域(亏格大于 0)．

2 本文方法概述

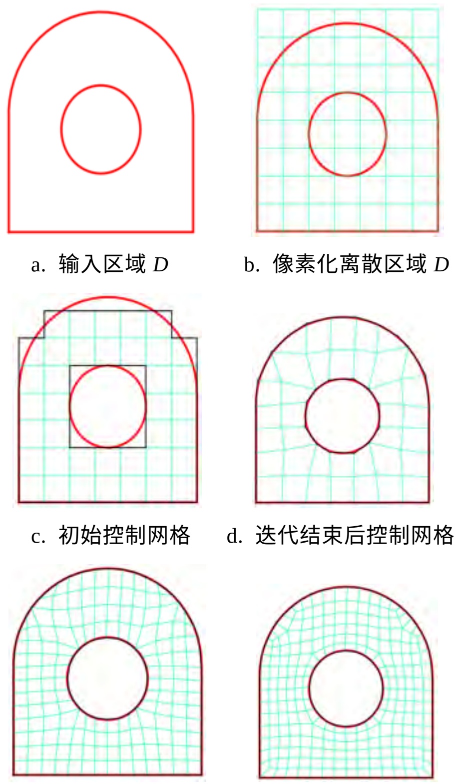
给定一条平面闭曲线，它围成一个平面区域 D (如图 1a 所示)，本文用几何迭代算法生成四边网格去填充这个区域 D．几何迭代法的主要步骤如下:

Step1．初始网格的生成．由主成分分析[26]构成区域D的包围盒，通过像素化将其离散，如图1b所示；然后利用扫描转化方法[27]处理，生成了一个均匀四边网格．这个四边网格经自适应剖分，得到了如图1c所示的初始四边网格，将它的边界多边形记为Γ_p．

Step2．边界曲线拟合．将有限步细分后的曲线当作拟合的曲线，将边界多边形Γ_p作为初始控制多边形，利用几何迭代对区域D的边界曲线进行拟合，当达到拟合精度时结束迭代，如图1d所示．

Step3．边界顶点移动的扩散．每次对边界多边形Γ_p迭代后，边界多边形Γ_p的顶点的移动通过分层Laplace光滑算子扩散到四边网格的内部顶点．

Step4．网格优化．几何迭代法停止后，对生成的四边网格的边界层进行 pillowing 操作[28]，并利用Mesquite[29]函数优化网格质量，如图1f所示．

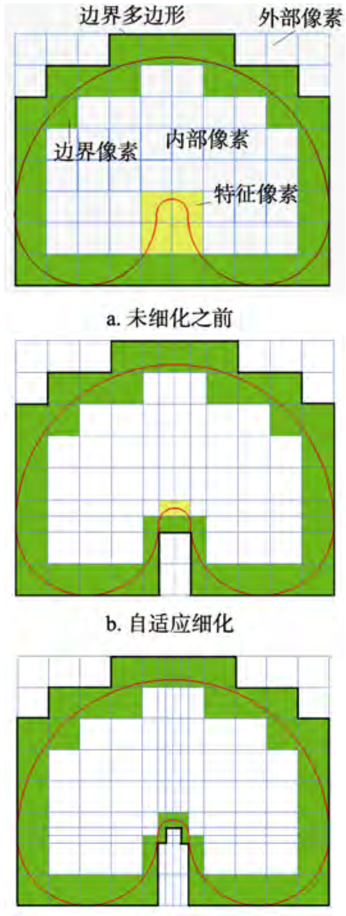


a. 输入区域 D　b. 像素化离散区域 D

c. 初始控制网格　d. 迭代结束后控制网格

e. Catmull-Clark 细分　f. 优化后网格

图 1　生成四边网格的算法示意图



a. 未细化之前

b. 自适应细化

c. 细化后

图 2　像素的自适应细化

3　几何迭代算法

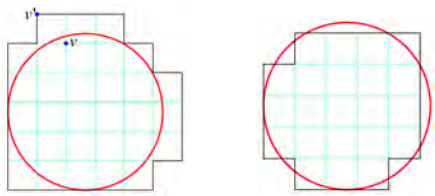
3.1　初始四边网格构造

给定一个如图 2a 所示的平面闭曲线 Γ, 它包围平面区域 D. 由主成分分析[26]构造区域 D 的包围盒, 这个包围盒被均匀像素化离散成四边网格. 为了生成理想的结果, 对不同的例子设置了不同的像素尺寸.

如图 2a 所示, 像素化将区域 D 离散成像素集合. 与曲线 Γ 相交的像素称为特征像素, 在区域 D 内部的像素称为内部像素, 在区域 D 外部的像素称为外部像素. 在特征像素的边中, 那些与一个外部像素和一个特征像素相邻的边, 或者只和特征像素相邻的边, 相互连接成边界多边形 Γ_p. 若某像素的边或者顶点在边界多边形 Γ_p 上, 则称该像素为边界像素. 注意到, 边界像素一定是特征像素, 但特征像素不一定是边界像素, 如图 2a 所示.

在图 2a 中, 有些像素(黄色)是特征像素但不是边界像素, 这使得边界多边形 Γ_p 的形状与给定的曲线 Γ 差异较大. 为了使二者的形状接近, 将非边界像素的特征像素(黄色)自适应细化(如图 2a 所示), 直到所有特征像素都变成边界像素(绿色)(如图 2c 所示). 这样, 一个包围区域 D 的四边网格就构造出来了.

此外, 这个四边网格的边界多边形 Γ_p 需要进一步调整, 使得它更接近于给定的曲线 Γ, 如图 3 所示. 具体来说, 考虑只有一个顶点在区域 D 内, 其他 3 个顶点在区域 D 外的边界像素, 如图 3a 所示, 在区域 D 内的顶点记为 v, 在区域 D 外的顶点记为 v'. 从这样的一个边界像素开始(如图 3a 所示), 如果 v 到曲线 Γ 的距离比 v' 到曲线 Γ 的距离近, 则删除这个边界像素; 否则, 将其保留, 如图



a. 边界调整前　b. 边界调整后

图 3　调整边界多边形 Γ_p

3a 所示. 这个调整操作迭代进行, 直到没有边界像素被删除为止. 最后, 将这样生成的四边网格作为初始四边网格, 它的边界多边形作为初始边界多边形 $\varGamma_p$, 用来拟合给定的曲线 $\varGamma$, 如图 3b 所示.

3.2 边界曲线拟合

本文中给定的曲线 $\varGamma$ 可以是任何形式, 如一个点列、一条折线段, 或者一条解析曲线. 如果曲线 $\varGamma$ 是一条折线段或者解析曲线, 从它上面采样, 得到点列

$$\{Q_i, i = 0,1,\cdots,m\} \tag{1}$$

则该点列将会由边界多边形 $\varGamma_p$ 拟合.

3.2.1 细分规则

在边界曲线拟合算法中, 拟合曲线取作边界多边形 $\varGamma_p$ 经过有限次细分之后生成的曲线, 由此, 填充平面 n 边域的四边网格也要通过细分控制四边网格同样次数之后得到. 特别地, 本文采用修改过的 Catmull-Clark 细分规则[30]. 下面解释对边界多边形 $\varGamma_p$ 进行细分的规则.

在执行细分之前, 边界多边形 $\varGamma_p$ 的顶点根据给定曲线 $\varGamma$ 的光滑性进行分类, 如图 4 所示. 如果 $\varGamma$ 是一条光滑曲线, 则多边形 $\varGamma_p$ 的所有顶点定义为光滑点; 否则, 如果曲线 $\varGamma$ 有尖点 p, 则多边形 $\varGamma_p$ 上最接近于 p 的点 p', 被定义为 $\varGamma_p$ 上的尖点, 其他点定义为光滑点.

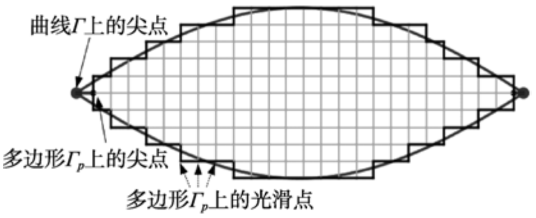


图 4 $\varGamma$ 上尖点和 $\varGamma_p$ 上尖点之间的对应

如图 4 所示, $\varGamma_p$ 上的尖点和 $\varGamma$ 上的尖点分别将 $\varGamma_p$ 和 $\varGamma$ 分成片段. $\varGamma_p$ 上的尖点和 $\varGamma$ 上的尖点之间的对应关系使得 $\varGamma_p$ 和 $\varGamma$ 上的片段相互对应, 且在接下来的迭代过程中被固定.

边界多边形 $\varGamma_p$ 细分生成 2 种新的顶点: 一种是边点 $E_{i,\mathrm{new}}$, 与 $\varGamma_p$ 的每一条边相对应; 另一种是顶点点 $P_{i,\mathrm{new}}$, 与 $\varGamma_p$ 的每个顶点对应.

由图 5 可知, 每个边点的生成方式为

$$E_{i,\mathrm{new}} = \frac{1}{2}\left(P_{i,\mathrm{old}} + P_{i-1,\mathrm{old}}\right).$$

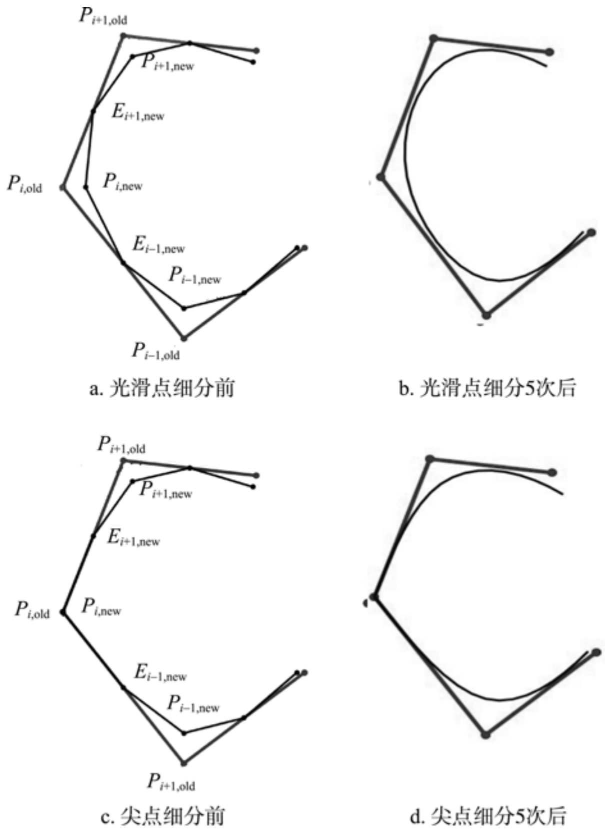


图 5 光滑点和尖点的细分规则

如果 $P_{i,\mathrm{old}}$ 是光滑顶点(如图 5a 所示), 与之相关的新顶点由

$$P_{i,\mathrm{new}} = \frac{1}{4}\left(E_{i,\mathrm{new}} + E_{i-1,\mathrm{new}} + 2P_{i,\mathrm{old}}\right)$$

生成. 另外, 如果 $P_{i,\mathrm{old}}$ 是一个尖点(如图 5c 所示), 新顶点

$$P_{i,\mathrm{new}} = P_{i,\mathrm{old}}.$$

上述细分规则中的所有下标均为边界多边形顶点个数取模后得到的.

3.2.2 几何迭代算法及其收敛性

本节介绍几何迭代算法, 并证明其收敛性.

把初始边界多边形 $\varGamma_p$ 作为初始控制多边形, 即 $\varGamma^{(0)} = \varGamma_p$, 并假设当几何迭代算法迭代到第 k 次时, 第 k 次控制多边形为 $\varGamma^{(k)}$, 控制顶点为 $\{P_i^{(k)},\ i = 0,1,\cdots,n\}$.

生成第 $(k+1)$ 控制多边形 $\varGamma^{(k+1)}$ 的迭代步骤包含矢量分配和矢量平均 2 个过程, 如图 6 所示. 下面分别阐述.

矢量分配. 记控制多边形 $\varGamma^{(k)}$ 第 l 次细分之后的多边形为 $\gamma_l^{(k)}$, 顶点为 $\{G_i^{(k)}\}$, 如图 6 所示. 基于

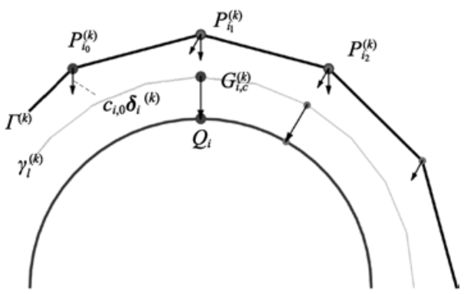


图 6　从 $\Gamma^{(k)}$ 生成 $\Gamma^{(k+1)}$ 的迭代步骤

细分规则可知，$\gamma_l^{(k)}$ 的每个顶点 $G_i^{(k)}$ 是控制多边形 $\Gamma^{(k)}$ 的一些控制顶点的线性组合，即

$$G_i^{(k)} = c_{i,0}P_{i_0}^{(k)} + c_{i,1}P_{i_1}^{(k)} + \cdots + c_{i,d}P_{i_d}^{(k)} \tag{2}$$

其中，$c_{i,1} > 0, c_{i,2} > 0, \cdots, c_{i,d} > 0$，并且 $\sum_{j=0}^{d} c_{i,j} = 1$.

对于式(1)中每个给定的数据点 Q_i，根据点的位置坐标和此点处的法向量坐标组成的 4 维坐标向量 (x_i, y_i, x_i^n, y_i^n)，计算点 Q_i 到 $\gamma_l^{(k)}$ 的距离最近的点 $G_{i,c}^{(k)}$，其中 (x_i, y_i) 为点的位置坐标，$\left(x_i^n, y_i^n\right)$ 为此点处的法向量坐标. 对应数据点 Q_i 的差向量 $\boldsymbol{\delta}_i^{(k)}$ 由

$$\boldsymbol{\delta}_i^{(k)} = Q_i - G_{i,c}^{(k)}, k = 0,1,\cdots \tag{3}$$

生成，如图 6 所示. 接着，将差向量 $\boldsymbol{\delta}_i^{(k)}$ 分配到控制多边形 $\Gamma^{(k)}$ 上的相关控制顶点上，这些控制顶点按照式(2)生成 $G_{i,c}$. 具体来说，每一个这样的控制顶点按照公式(2)中的系数被分配一个加权矢量. 例如，分配到控制顶点 $P_{i_j}^{(k)}$ 的加权矢量为 $c_{i,j}\boldsymbol{\delta}_i^{(k)}$，如图 6 所示.

矢量平均. 当矢量分配完成之后，控制多边形的每个控制顶点 $P_j^{(k)}$ 被分配到的加权矢量集合为 $\left\{c_{i,j}\boldsymbol{\delta}_i^{(k)}, i = 0,1,\cdots,e\right\}$，如图 6 所示. 对这些矢量进行加权平均操作，产生了对应于控制顶点 $P_j^{(k)}$ 的差向量

$$\boldsymbol{\varDelta}_j^{(k)} = \frac{\sum_{i=0}^{e} c_{i,j}\boldsymbol{\delta}_i^{(k)}}{\sum_{i=0}^{e} c_{i,j}} \tag{4}$$

此外，如果 $P_j^{(k)}$ 是光滑顶点，在 $P_j^{(k)}$ 上加上 $\boldsymbol{\varDelta}_j^{(k)}$ 生成新的控制多边形 $\Gamma^{(k+1)}$ 上的控制顶点 $P_j^{(k+1)}$，

$$P_j^{(k+1)} = P_j^{(k)} + \boldsymbol{\varDelta}_j^{(k)},\quad k = 0,1,\cdots,n;$$

如果 $P_j^{(k)}$ 是尖点，则令

$$P_j^{(k+1)} = P_j^{(k)}.$$

下面证明上述几何迭代法的收敛性.

将对应控制点的差向量 $\boldsymbol{\varDelta}_j^{(k)}$ 排成一个序列

$$\boldsymbol{D}^{(k)} = [\boldsymbol{\varDelta}_0^{(k)}, \boldsymbol{\varDelta}_1^{(k)}, \cdots, \boldsymbol{\varDelta}_n^{(k)}]^{\mathrm{T}}, k = 0,1,\cdots$$

根据式(4)，有

$$\begin{aligned}
\boldsymbol{\varDelta}_j^{(k+1)} &= \frac{\sum_{i=0}^{e} c_{i,j}\boldsymbol{\delta}_i^{(k+1)}}{\sum_{i=0}^{e} c_{i,j}} = \\
&\frac{\sum_{i=0}^{e} c_{i,j}\left(Q_i - \sum_l c_{i,l}\left(P_{i_l}^{(k)} + \boldsymbol{\varDelta}_{i_l}^{(k)}\right)\right)}{\sum_{i=0}^{e} c_{i,j}} = \\
&\frac{\sum_{i=0}^{e} c_{i,j}\boldsymbol{\delta}_i^{(k)}}{\sum_{i=0}^{e} c_{i,j}} - \frac{\sum_{i=0}^{e}\sum_l c_{i,l}\boldsymbol{\varDelta}_{i_l}^{(k)}}{\sum_{i=0}^{e} c_{i,j}} = \\
&\boldsymbol{\varDelta}_j^{(k)} - \frac{\sum_{i=0}^{e}\sum_l c_{i,l}\boldsymbol{\varDelta}_{i_l}^{(k)}}{\sum_{i=0}^{e} c_{i,j}}.
\end{aligned}$$

由此，得到几何迭代格式的矩阵形式

$$\boldsymbol{D}^{(k+1)} = \left(\boldsymbol{I} - \boldsymbol{\varLambda}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)\boldsymbol{D}^{(k)} \tag{5}$$

其中，

$$\boldsymbol{\varLambda} = \mathrm{diag}\left\{\frac{1}{\sum_{i\in I_0} c_{i,0}}, \frac{1}{\sum_{i\in I_0} c_{i,1}}, \cdots, \frac{1}{\sum_{i\in I_0} c_{i,n}}\right\},$$

$$\boldsymbol{A} = \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,n} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,0} & c_{n,1} & \cdots & c_{n,n} \end{bmatrix};$$

并且 $\boldsymbol{I}_j$，$j = 0,1,\cdots,n$ 是向第 j 个控制顶点分配差向量的数据点的指标集合.

注意到 $\boldsymbol{\varLambda}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 是对称矩阵. 如果 $\boldsymbol{A}$ 满秩，则 $\boldsymbol{\varLambda}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 是一定正定的. 所以，它的特征值都是正实数，即

$$\lambda\left(\boldsymbol{\varLambda}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right) > 0.$$

另一方面，由于 $\left\|\boldsymbol{\varLambda}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right\|_\infty = 1$，则

$$\lambda\left(\boldsymbol{\varLambda}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right) \leqslant 1.$$

因此，$0 < \lambda\left(\boldsymbol{\varLambda}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right) \leqslant 1$. 矩阵 $\boldsymbol{I} - \boldsymbol{\varLambda}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的特征值

非负且小于 1，即

$$0 \leqslant \lambda\left(\boldsymbol{I}-\Lambda \boldsymbol{A}^{\mathrm{T}} \boldsymbol{A}\right)=1-\lambda\left(\Lambda \boldsymbol{A}^{\mathrm{T}} \boldsymbol{A}\right)<1 .$$

所以，$\boldsymbol{I}-\Lambda \boldsymbol{A}^{\mathrm{T}} \boldsymbol{A}$ 的谱半径小于 1. 说明式(5)所示的几何迭代格式收敛.

本文实验中，当

$$\left|\frac{\sum_{i=0}^{m}\left\|\boldsymbol{\delta}_i^{(k+1)}\right\|}{\sum_{i=0}^{m}\left\|\boldsymbol{\delta}_i^{(k)}\right\|}-1\right|<\varepsilon$$

时，几何迭代停止，其中 $\varepsilon=0.05$.

3.3 内部顶点的移动

在第 3.2.2 节中，几何迭代每一步结束之后，边界控制顶点的改变会导致四边控制网格内部顶点的改变，这个任务是通过一个分层 Laplace 算子来完成的.

1) 四边控制网格的内部顶点分成很多层. 内部点的第一层由那些与边界控制顶点相邻的内部顶点组成；第二层由那些与第一层内部顶点相邻的内部顶点组成，等.

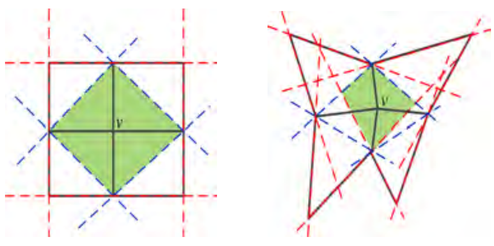
2) 按照由外到内的顺序，一层一层地对这些顶点进行 Laplace 算子操作，内部顶点 v 移动到新顶点位置 v_{new} ,

$$v_{\text{new}}=\frac{\sum_{j=1}^{d} v_j}{d} .$$

其中，d 是顶点 v 的度，v_j ($j=1,2,\cdots,d$)是 v 的一环邻域点. 上述内部顶点移动的扩散算法迭代进行，直到顶点的最大移动距离小于一个给定值为止. 本文实现中，该值取为 0.05. 这样，边界控制顶点的改变扩散到了内部顶点.

3.4 质量保证

上述几何迭代法从一个初始四边网格开始，其中所有四边形都是矩形，所有角都是直角，如图 7a 所示. 因此，初始四边网格保证严格不自交. 在迭代过程中，保证每次迭代生成的四边网格不自交，等价于使每个四边形的每个角小于 π . 假设第 k 次迭代得到的四边网格是严格不自交的. 如图 7 所示，称包含顶点 v 的绿色区域为可行域，它是由与 v 相邻的四边形的对角线(图 7 中的蓝色虚线)，以及与 v 相邻的四边形的所有边中，顶点 v 的一环领域点相邻接的边决定的直线(图 7 中的红色虚线)所包围的区域. 在第 $(k+1)$ 次迭代后，如果顶点 v 没有超出上述可行域，与顶点 v 邻接的四边形的角将会保持小于 π . 这样，保证了生成的四边网格是不自交的.



a. 初始网格的可行域　　b. 迭代后网格的可行域

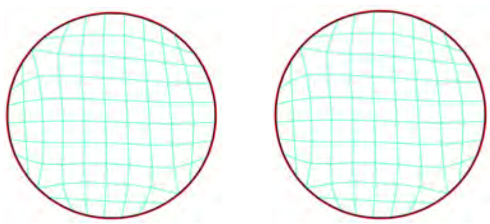
图 7　网格顶点的可行域

因此，如果由顶点 v 加上式(4)所示的差向量生成的顶点，仍然位于顶点 v 的可行域内，则将它取作新顶点；但是，如果它超出了可行域，本文把式(4)所示的差向量等分成 10 段，选取在可行域内，且离可行域边界最近的分点作为新顶点，如图 7 所示.

在每次边界顶点移动，以及边界顶点移动扩散到内部顶点的过程中，顶点的移动严格保持在可行域中. 该策略保证了生成的四边网格是严格不自交的.

3.5 网格质量优化

如图 8 所示，几何迭代结束之后，经过对控制网格有限次细分生成了填充给定区域 D 的四边网格. 在这样生成的四边网格中(如图 8a 所示)，质量较差的网格单元集中在边界层周围. 利用第 3.3 节的分层 Laplace 算法，边界层附近较差的网格质量对内部顶点的影响越来越小，内部顶点距离边界层越远，影响越小. 这使得所生成四边网格在大多数内部网格顶点处是正交的，或接近正交的，如图 8 所示.



a. 优化之前　　b. 优化之后

图 8　网格质量优化

为了提高边界层附近的网格质量，本文使用 pillowing 算法[28]，在第一层和边界层之间插入新的一层网格顶点，如图 8b 所示. 最后，四边网格的质量通过 Mesquite 库函数进行优化[29]. 图 8b 所示为在 pillowing 和 Mesquite 优化之后的网格.

4　实验结果与讨论

本文提出的四边网格生成算法已经在 Visual C++开发环境下实现，程序运行环境为 Intel Core2 Quad CPU Q9400 2.66 GHz 和 4 GB 内存.

本节展示四边网格生成算法的一些实验结果. 本文实验中，拟合精度定义为 RMS 误差，与给定 n 边域包围盒对角线长度的比值，即

$$\frac{\sqrt{\dfrac{\sum_{i=0}^{m}\left\|\boldsymbol{\delta}_i^{(k)}\right\|^2}{m+1}}}{L}.$$

其中，$\boldsymbol{\delta}_i^{(k)}$ 是几何迭代结束后对应数据点的差向量，其定义同式(3)；$m+1$ 是数据点的个数，其定义同式(1)；L 是 n 边域包围盒对角线长度.

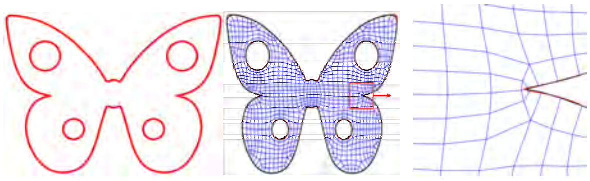
为了衡量四边网格的质量，本文采用归一化雅克比行列式[31]，其定义如下：给定一个顶点逆时针排列的四边形 $ABCD$，每个顶点对应一个归一化雅克比行列式. 记 $\boldsymbol{e}_{ab}=(x_{ab},y_{ab})$ 和 $\boldsymbol{e}_{ad}=(x_{ad},y_{ad})$ 分别为 $\boldsymbol{AB}$ 和 $\boldsymbol{AD}$ 的单位矢量，则顶点 $\boldsymbol{A}$ 对应的归一化雅克比行列式定义为

$$J=\begin{vmatrix} x_{ab} & y_{ab} \\ x_{ad} & y_{ad} \end{vmatrix}.$$

其他顶点对应的归一化雅克比行列式可类似定义. 把四边形的四个顶点对应的归一化雅克比行列式值中最小的作为这个四边形的归一化雅克比行列式值. 显然，四边网格越接近于正交，雅克比行列式值就越大. 如果顶点 $\boldsymbol{A}$ 的角是直角，此时顶点 $\boldsymbol{A}$ 对应的雅克比行列式值为 1.

图 9~11 展示了本文方法生成的一些四边网格，这些网格都是经过一次 Cutmull-Clark 细分之后生成的，它们的统计数据如表 1 所示. 在本文方法的实现中，像素化的初始网格尺寸是通过将给定区域 D 的包围盒的最小边长度乘以一个分数值得到的(见第 3.1 节)，在表 1 最后一列列出了这些分数值.

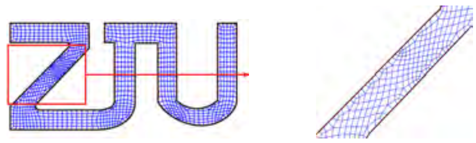
图 9 中，本文方法被用来填充具有复杂形状和复杂拓扑结构的平面区域. 图 9a 所示为含有 4 个洞的平面区域；生成的四边网格如图 9b 所示，其中四边网格的边界用黑色曲线表示，给定的平面区域的边界用红色曲线表示. 在有限元分析中，一个可用的四边网格的雅克比行列式的值应该在 [0.2,1]. 从表 1 可以看出，所生成的四边网格的雅克比行列式的最小值和平均值分别是 0.262 5 和 0.943 4.



a. 给定的平面域边界　b. 生成的四边网格　c. 部分网格放大图

图 9　在复杂形状和复杂拓扑结构的平面域上生成四边网格

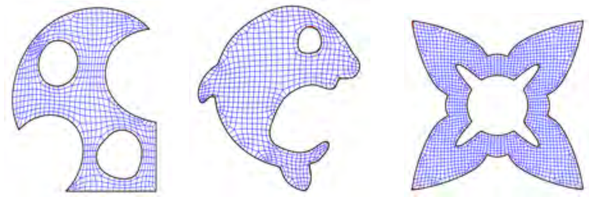
a. 给定的平面域边界



b. 生成的四边网格　　c. 部分网格放大图

图 10　在空心字母“ZJU”内部生成四边网格

a. 给定的平面域 1　b. 给定的平面域 2　c. 给定的平面域 3



d. 生成的四边网格 1 e. 生成的四边网格 2　f. 生成的四边网格 3

图 11　四边网格生成的其他例子

图 10 中，本文方法生成的四边网格被用来填充空心字母“ZJU”，空心字母的边界由一些直线和弧线拼接而成. 可以看出，虽然空心字母包围的平面区域的形状非常复杂(如图 10a 所示)，本文方法生成的四边网格(如图 10b 所示)还是具有理想的质量. 雅克比行列式的平均值和最小值分别是 0.954 0 和 0.505 7(如表 1 所示).

图 11 展示了另外 3 个例子. 在这 3 个例子中，本文方法生成的四边网格都有不错的质量，如表 1 所示.

此外，我们还比较了本文方法和文献[19]中

表 1　几何迭代法生成四边网格的统计数据

图序号	顶点数	四边形数	精度	时间/s	平均雅克比值	最小雅克比值	初始尺寸
9b	1063	936	0.0044	21.39	0.9434	0.2625	1/12
10b	1209	1040	0.0077	21.312	0.9540	0.5057	1/10
11d	709	620	0.0092	19.55	0.9271	0.3407	1/14
11e	740	652	0.0040	16.18	0.9533	0.5847	1/10
11f	1744	1552	0.0019	45.35	0.9645	0.5717	1/10

提出的方法，如图 12 所示．通过表 2 可以看出，本文方法生成的四边网格的质量(如图 12b 所示)，要优于文献[19]中方法生成的网格质量．

为进一步比较本文方法和文献[19]中方法生成的网格质量，本文在这 2 种方法生成的网格上，分别用有限元分析(双线性形函数)求解以下偏微分方程

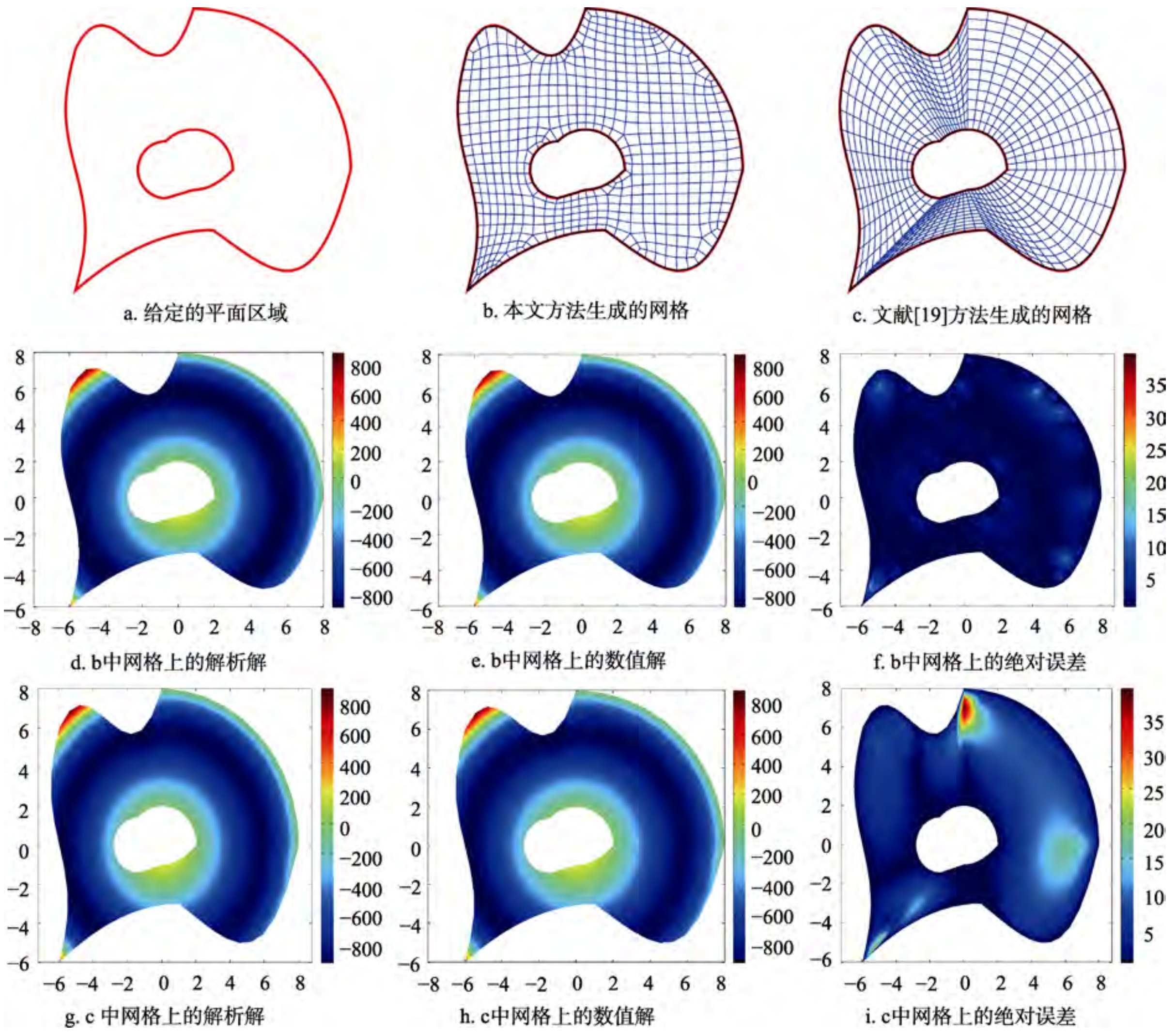


图 12　2 种方法生成的四边网格的比较

表 2　2 种方法生成的四边网格的数据比较

方法	顶点数	四边形数	平均雅克比值	最小雅克比值	L^2 误差	L^∞ 误差
本文	642	572	0.9649	0.2795	45.8791	13.5398
文献[19]	780	720	0.8982	0.0976	159.7548	39.3401

$$\begin{cases}\dfrac{\partial^2 u}{\partial x^2}+\dfrac{\partial^2 u}{\partial y^2}=16x^2+16y^2-272\\ u|_{\partial D}=\left(x^2+y^2-4\right)\left(x^2+y^2-64\right)\end{cases};$$

其中 $D=[-6,8]\times[-6,8]$，它的解析解为 $u(x,y)=(x^2+y^2-4)(x^2+y^2-64)$．

图 12d，12g 是相应网格上的解析解，图 12e，12h 是相应网格上的数值解，图 12f，12i 是相应网格上的绝对误差的分布，定义为

$$e(x,y)=|u(x,y)-\bar{u}(x,y)|,$$

其中，$u(x,y)$ 是解析解，$\bar{u}(x,y)$ 是数值解．

这个例子演示了网格质量对数值解的影响．如表 2 所示，本文方法生成四边网格(如图 12b 所示)

的质量优于文献[19]生成的四边网格(如图 12c 所示)的质量．虽然图 12c 中网格的四边形数比图 12b 中网格的四边形数要多，但本文方法生成的网格(如图 12b 所示)上数值解的精度要优于文献[19]生成的四边网格(如图 12c 所示)上数值解的精度．特别地，如表 2 所示，本文方法生成四边网格只有 572 个四边形(如图 12b 所示)，其上数值解的 L^2 误差和 L^∞ 误差分别是 45.879 1 和 13.539 8．另一方面，文献[19]方法生成的四边网格(如图 12c 所示)有 720 个四边形单元，但其上数值解的 L^2 误差和 L^∞ 误差分别是 159.754 8 和 39.340 1．

在本文方法的各个步骤中，最近点的计算是一个关键步骤，它实际上是一个点的匹配问题．一方面，在凹区域难以获得理想的匹配点；另一方面，计算最近点需要的计算量较大．因此，需要设计更为合理的最近点计算方法．此外，算法的计算效率可以从以下方面提高：1) 引入 K-D 树结构，提高最近点的计算速度；2) 在几何迭代法的每一步迭代中，引入并行机制．

5　结　语

本文提出一种基于几何迭代的四边网格生成算法，用于填充具有复杂形状和复杂拓扑的 $n(n\neq4)$ 边平面域．本文方法生成的四边网格能保证严格不自交．具体来说，首先通过像素化离散给定平面域得到的初始控制网格；然后逐步调整控制四边网格的边界拟合给定区域的边界曲线；在每一步控制网格边界顶点迭代调整后，边界顶点的移动通过分层 Laplace 算子向控制网格的内部网格顶点扩散．进一步，本文分析了几何迭代算法的收敛性．在有限次细分之后，四边网格就生成了．此外，四边网格质量通过 Pillowing 操作和 Mesquite 函数得到进一步优化．本文方法可以在复杂形状和复杂拓扑结构的平面域上生成严格不自交四边网格．大量实验结果表明，本文方法生成的四边网格的质量对于有限元分析是可取的．

参考文献(References):

[1] Wang C C L, Tang K. Non-self-overlapping structured grid generation on an n-sided surface[J]. International Journal for Numerical Methods in Fluids, 2004, 46(9): 961-982

[2] Owen S J, Staten M L, Canann S A, *et al.* Q-morph: an indirect approach to advancing front quad meshing[J]. International Journal for Numerical Methods in Engineering, 1999, 44(9): 1317-1340

[3] Lee Y K, Lee C K. Automatic generation of anisotropic quadrilateral meshes on three-dimensional surfaces using metric specifications[J]. International Journal for Numerical Methods in Engineering, 2002, 53(12): 2673-2700

[4] Tam T K H, Armstrong C G. 2D finite element mesh generation by medial axis subdivision[J]. Advances in Engineering Software and Workstations, 1991, 13(5/6): 313-324

[5] Yamakawa S, Shimada K. Quad-layer: layered quadrilateral meshing of narrow two-dimensional domains by bubble packing and chordal axis transformation[J]. Journal of Mechanical Design, 2002, 124(3): 564-573

[6] Thompson J F. A survey of dynamically-adaptive grids in the numerical solution of partial differential equations[J]. Applied Numerical Mathematics, 1985, 1(1): 3-27

[7] Eça L. 2D orthogonal grid generation with boundary point distribution control[J]. Journal of Computational Physics, 1996, 125(2): 440-453

[8] Soni B K. Elliptic grid generation system: control functions revisited[J]. Applied Mathematics and Computation, 1993, 59 (2/3): 151-163

[9] Khamayseh A, Hamann B. Elliptic grid generation using NURBS surfaces[J]. Computer Aided Geometric Design, 1996, 13(4): 369-386

[10] Knupp P M. Jacobian-weighted elliptic grid generation[J]. SIAM Journal on Scientific Computing, 1996, 17(6): 1475- 1490

[11] Kim S. Control functions and grid qualities measurements in the elliptic grid generation around arbitrary surfaces[J]. International Journal for Numerical Methods in Fluids, 2000, 33(1): 81-88

[12] Akcelik V, Jaramaz B, Ghattas O. Nearly orthogonal two-dimensional grid generation with aspect ratio control[J]. Journal of Computational Physics, 2001, 171(2): 805-821

[13] Zhang Y X, Jia Y F, Wang S S Y. 2D nearly orthogonal mesh generation with controls on distortion function[J]. Journal of Computational Physics, 2006, 218(2): 549-571

[14] Zhang Y X, Jia Y F, Wang S S Y. An improved nearly-orthogonal structured mesh generation system with smoothness control functions[J]. Journal of Computational Physics, 2012, 231(16): 5289-5305

[15] Smith R E, Eriksson L E. Algebraic grid generation[J]. Computer Methods in Applied Mechanics and Engineering, 1987, 64(1-3): 285-300

[16] Shih T I P, Bailey R T, Nguyen H L, *et al*. Algebraic grid generation for complex geometries[J]. International Journal for Numerical Methods in Fluids, 1991, 13(1): 1-31

[17] Brakhage K H, MÄuller S. Algebraic-hyperbolic grid generation with precise control of intersection of angles[J]. International Journal for Numerical Methods in Fluids, 2000, 33(1): 89-123

[18] Yang D C H, Chuang J J, Oulee T H. Boundary-conformed toolpath generation for trimmed free-form surfaces[J]. Computer-Aided Design, 2003, 35(2): 127-139

[19] Wang C C L, Tang K. Algebraic grid generation on trimmed parametric surface using non-self-overlapping planar coons patch[J]. International Journal for Numerical Methods in Engi-

neering, 2004, 60(7): 1259-1286

[20] Wang C C L, Tang K. Non-self-overlapping hermite interpolation mapping: a practical solution for structured quadrilateral meshing[J]. Computer-Aided Design, 2005, 37(2): 271-283

[21] Lin H W, Tang K, Joneja A, *et al*. Generating strictly non-self-overlapping structured quadrilateral grids [J]. Computer-Aided Design, 2007, 39(9): 709-718

[22] Thompson J F, Soni B K, Weatherill N P. Handbook of grid generations[M]. Boca Raton: CRC Press, 1999

[23] Baker T J. Mesh generation: art or science?[J]. Progress in Aerospace Sciences, 2005, 41(1): 29-63

[24] Mei Zhongyi, Fan Yuqing. A modified paving technique for quadrilateral mesh generation in planar regions[J]. Journal of Computer-Aided Design & Computer Graphics, 2000, 12(6): 428-434(in Chinese)
(梅中义，范玉青. 一种用于自动生成二维全四边形有限元网格的改进的铺设算法[J]. 计算机辅助设计与图形学学报, 2000, 12(6): 428-434)

[25] Li Yi, Bao Jinsong, Jin Ye, *et al*. Quadrilateral mesh generation algorithm for planar domain with multi-constraints[J]. Journal of Computer-Aided Design & Computer Graphics, 2008, 20(4): 488-493(in Chinese)
(李 毅, 鲍劲松, 金 烨, 等. 二维域多约束四边形有限元网格生成算法[J]. 计算机辅助设计与图形学学报，2008, 20(4), 488-493)

[26] Jolliffe I T. Principal component analysis[M]. Hoboken: Wiley, 2005

[27] Foley J, van Dam A, Feiner S, *et al*. Computer graphics in C#: principles and practices[M]. Boston: Addison-Wesley Publishing Company, 2008

[28] Tautgesa T J, Knoopb S E. Topology modification of hexahedral meshes using atomic dual-based operations[C] //Proceedings of the 12th International Meshing Roundtable. Sandia: Sandia National Laboratories, 2003: 415-423

[29] Brewer M, Diachin L F, Knupp P, *et al*. The mesquite mesh quality improvement toolkit[C] //Proceedings of the 12th International Meshing Roundtable. Sandia: Sandia National Laboratories, 2003: 239-250

[30] Catmull E, Clark J. Recursively generated B-spline surfaces on arbitrary topological meshes[J]. Computer-Aided Design, 1978: 10(6): 350-355

[31] Knupp P M. A method for hexahedral mesh shape optimization[J]. International Journal for Numerical Methods in Engineering, 2003, 58(2): 319-332