中国科学技术大学

学士学位论文

多正多边形辅助的 2 维区域四边形网格化

姓　　名：拜 重 阳

院　　系：数学科学学院信息与计算科学系

学　　号：PB12001087

导　　师：刘洋

完成时间：二〇一六年六月

University of Science and Technology of China

A dissertation for bachelor's degree

polyquad

Author :	Chongyang Bai
Department :	
Student ID :	PB12001087
Supervisor :	Dr. Yang Liu
Finished Time :	June, 2016

致 谢

拜重阳
2016 年 5 月 19 日

目　录

表格索引

插图索引

算法索引

摘 要

在科学研究、工程计算、文化娱乐中，数字几何数据扮演着越来越重要的角色。使用数学模型和算法来分析与处理数字几何数据的过程称作数字几何处理。这是一个包含计算机科学、应用数学和工程学等学科的交叉性研究课题。常见的研究内容包括模型获取、模型重建、网格生成、形状分析与理解、映射计算和几何建模等。我们的研究针对数字几何处理中的两个子课题：最优映射计算和最优网格生成。其中最优映射计算是一个重要的课题，它是许多计算机图形学应用的核心，比如网格参数化、网格变形、网格质量提高、六面体网格生成。最优网格生成是网格数据处理的基石，比如各向异性的网格和六面体网格有很强的需求，因为在有限元方法中，它们能获得比各向性网格和四面体网格更好的计算精度。最优映射计算可以作为网格生成的后处理技术，使生成的网格质量得到提高。

本文从优化的角度设计了新颖的能量函数和优化方法，将它们成功地应用到了最优网格映射计算、各向异性网格生成和多立方体结构 (polycube) 自动生成这几个课题，具体如下：

一个好的映射算法需要保证无翻转、低形变和计算高效性。现有的算法不能同时保证这些特性。本文设计了一个新颖的形变最小化能量 (Advanced Most-Isometric ParameterizationS，AMIPS)，并使用非精确块坐标轮换下降算法来快速地计算无网格翻转的最优映射。其中 AMIPS 能量函数继承了传统的 MIPS 能量的保证无翻转的性质，同时能控制最大的形变。非精确块坐标轮换下降算法 (inexact Block Coordinate Descent，inexact BCD) 能避免优化过程过早地陷入局部最小。结合 AMIPS 能量函数与 inexact BCD 优化算法，本文提高了映射的计算效率和质量。在网格参数化、二维三角形网格与三维四面体网格变形、二维与三维无网格变形、各向异性四面体和六面体网格质量提高等应用中充分体现了我们算法的优越性。

但是 AMIPS 算法同样存在缺点：比如不能支持存在很多控制点的网格变形，而且对初始映射比较敏感。本文提出了一个网格组装分离方法来计算无翻转的最优映射。我们的方法接受任意的网格映射作为输入，该输入映射可以存在众多翻转的网格单元。我们首先将网格的所有网格单元分离，保持每个网格单元上的映射是低形变的，然后通过同时优化形变和分离顶点之间的距离来计算无翻转的最优映射。由于使用了每个网格单元上的仿射变换作为优化变量，我们可以通过求解一个无约束的非线性非凸优化问题来得到最优映射。同样在平面网格参数化、网格变形等应用中体现了我们算法的鲁棒性和高效性。

在几何建模、物理模拟和机械工程等应用中，各向异性网格是非常重要的。本文提出了局部凸函数三角化 (Local Convex Triangulation，LCT) 方法，用于生成高质量的各向异性网格。输入一个曲面，或者一个三维空间区域作为定义域，

和在定义域上的已知黎曼度量场，我们将各向异性网格生成问题转化为一个函数逼近问题。在每个网格单元上构造局部凸函数，它的 Hessian 矩阵局部上和输入的黎曼度量一致，我们利用网格顶点移动和改变网格连接关系的策略来降低函数逼近误差。我们的 LCT 方法推广了最优 Dealunay 三角化 (Optimal Delaunay Triangulation，ODT)，可以接受黎曼度量场作为输入，同时可以适用于剧烈变化的黎曼度量场和存在尖锐特征的网格。从二维平面区域、三维空间区域和三维曲面上生成的各向异性网格来看，我们算法效率高，结果网格质量高。

在物理模拟和机械工程等应用中，六面体网格往往比四面体网格有着较好的性质，比如更少的网格单元、更高的计算精度。本文通过高质量多立方体 (polycube) 结构来生成六面体网格。多立方体结构要求网格的表面三角形的法向和 X，Y，Z 轴严格对齐。之前的算法不能同时保证无翻转、低形变、奇异性可控和计算高效。本文使用非精确块坐标轮换下降算法来优化表面法向光滑与对齐能量，用来驱动网格变形并自动地消除极限点，以自动生成高质量的多立方体结构。我们引入光滑函数的核宽度来控制多立方体结构的奇异性。非精确块坐标轮换下降算法的高效率使本文的自动化算法的效率远远高于现在最先进的算法。从多立方体映射的形变和六面体网格生成的结果来看，我们算法的质量和效率相对于当前最先进的算法都有较大提升。

关键词： 数字几何处理 各向异性网格生成 局部凸函数网格化 无翻转映射 多立方体结构 形体变形 网格参数化

ABSTRACT

3D digital contents play important roles in scientific research, mechanical engineering, and entertainment. Digital geometric processing is a field that utilizes mathematical models and algorithms to analyze and manipulate geometric data. It is an interdisciplinary research relating to computer graphics, applied mathematics, and engineering. Typical geometric processing tasks include mesh acquisition, mesh reconstruction, mesh generation, shape analysis and understanding, mapping computation and geometric modeling. Our research focuses on two topics: optimal mapping computation and optimal mesh generation. Optimal mapping computation is a fundamental task in computer graphics, and it is the key to many applications, e.g. mesh parameterizations, mesh deformation, mesh improvement, all-hex mesh generation. Optimal mesh generation is one of the bases of digital geometric processing. For example, anisotropic and all-hex meshes are in great need, because they can provide more accuracy than isotropic and tetrahedral meshes in some applications. Optimal mapping can be used to improve the result of mesh generation.

In the thesis we design novel energy functions and optimization algorithms from the perspective of optimal theory and apply them to solve optimal mapping computation, anisotropic mesh generation, and PolyCube structure construction:

A good mapping possesses nice properties: inversion-free and low-distortion and its computation need to be efficient. State-of-the-art methods cannot satisfy all requirements. By revisiting the well-known MIPS (Most-Isometric ParameterizationS) method, we introduce an advanced MIPS(AMIPS) method that inherits the local injectivity of MIPS, achieves as low as possible distortions compared to the state-of-the-art locally injective mapping techniques, and performs one to two orders of magnitude faster in computing a mesh-based mapping. The success of our method relies on two key components. The first one is an enhanced MIPS energy function that penalizes the maximal distortion significantly and distributes the distortion evenly over the domain for both mesh-based and meshless mappings. The second is a use of the inexact block coordinate descent method in mesh-based mapping in a way that efficiently minimizes the distortion with the capability not to be trapped early by the local minimum. We demonstrate the capability and superiority of our method in various applications including mesh parameterizations, deformation, and mesh improvement.

AMIPS has some limitations: it cannot support many handles in mesh deformation and is sensitive to initial mappings. We present a novel method to compute locally injective mappings with low distortion on simplicial meshes. Given an initial mapping

with or without inverted simplices, our method first disassembles it into disjointed and inversion-free simplices by modifying the piecewise affine transformations defined on simplices, then minimizes the mapping distortion and the difference of the disjointed vertices with respect to the piecewise affine transformations. Due to the use of transformations as unknowns, our algorithm explicitly guarantees the local injectivity of the mapping via an unconstrained minimization. Compared with existing methods, our method is robust to initial mappings even with many inverted elements and handle constraints. Our method is also capable of achieving bounded distortion mappings. We demonstrate the efficiency and robustness of our method on a variety of applications.

Anisotropic meshes are very important in geometric modeling, physical simulation and mechanical engineering. We present a novel approach for high-quality anisotropic triangle mesh generation, called local convex triangulation(LCT). Given a 2D or surface domain equipped with Riemannian metrics, the anisotropic meshing is transformed into a functional approximation problem. We construct convex functions locally over the mesh to best match Riemannian metrics, and adapt vertex positions and mesh connectivity to minimize the interpolation error iteratively to achieve the desired anisotropy. We show that our method is a generalization of Optimal Delaunay Triangulation, and we develop a simple and efficient algorithm that works well for 2D and surface meshing. The superiority of our method in mesh quality and algorithmic efficiency in comparison to existing methods is demonstrated with a variety of models and Riemannian metrics.

All-hex meshes possess nice numerical properties, such as a reduced number of elements and high approximation accuracy in physical simulation and mechanical engineering. We generate high-quality all-hex meshes based on PolyCubes which are good abstractions of closed shapes. A desired PolyCube construction method should (1) provide a low-distortion map without foldover and degeneracy; (2) offer flexible control on singularity counts; (3) compute the result efficiently and automatically. We introduce a novel method that fulfills these requirements to compute PolyCubes for tetrahedral meshes. We regard the computation as mesh deformation that is driven by face normal smoothing and axis-directional alignment under distortion control. The kernel size of our Gaussian smoothing is used to control the singularity count, and the level of alignment is adjusted automatically to resolve the turning point issue. We formulate the deformation as an optimization problem and propose a very efficient solver. We demonstrate the quality and speed of our method compared to state-of-the-art methods on a variety of models.

Keywords: Digital geometric processing, Anisotropic meshing, Local convex triangulation, Inversion-free mappings, PolyCube, Parameterizations, Deformation

第一章 绪论

本章首先回顾了 2 维四边形网格的生成及应用，然后简单介绍了无翻转映射这一概念。本文正是把无翻转的映射作为输入区域和简化区域——多正多边形结构的桥梁，在多正多边形结构上生成四边形网格后，再映回原始区域得到需要的四边形网格。

1.1 2 维四边形网格生成技术

1.1.1 四边形网格的应用

许多工程上的问题都归结为在连续的物理系统中求解偏微分方程来进行分析和模拟。比如模拟飞机在不同的气流环境中的飞行表现、计算集成电路中的电场分布、利用波动方程分析油气储集层的地质情况等。由于很多偏微分方程难以找到解析解，人们利用计算机模拟这些系统，建立离散的模型，就可以求出一定误差内的数值解。为了在这些离散的点组成的空间或平面上利用点的邻域近似地计算导数，我们需要把这些连续的定义域离散化为网格。图展示了平面和空间中常见的网格模型。

有限元方法是解不规则区域上的偏微分方程的有力技术。平面上的区域被离散化为尽量规则的三角形或四边形网格，而空间区域被离散化为四面体或六面体网格。一般而言，四边形网格具有更高的计算精度和更快的收敛速度。虽然计算机运算速度的飞速发展使得计算数值解越来越快，但生成复杂区域的高质量四边形网格仍然要花费大量时间，甚至需要提供繁琐而艰辛的用户交互。全自动、快速、鲁棒地生成高质量的四边形网格成为了进行有限元分析的重要前提。

1.1.2 四边形网格的质量

有限元方法对输入的四边形网格有如下几个要求：

无翻转四边形 无效的翻转单元会严重影响数值解的精度以及收敛。我们给出

定义 1.1.1. *把四边形的顶点按逆时针顺序记为 v_1, v_2, v_3, v_4, 考虑三角形 $\triangle v_1v_2v_3, \triangle v_2v_3v_4, \triangle v_3v$ 若它们的有向面积均有至少一个为负，则称四边形是翻转的, 我们称没有翻转四边形的网格是有效网格。图（分 4 个三角形，翻转）*

接近原始边界 离散化的四边形网格在边界上要和连续的边界曲线尽量接近，否则会导致数值解的误差过大而不能满足需求。

规则的四边形单元 越接近正方形或矩形的四边形越规则，规则的四边形单元会提高有限元方法的计算精度和速度。一个四边形网格质量的常用几何度量就是考察它的角度是否接近直角。

较少的奇异点 四边形网格中的奇异点会减慢解方程的速度，并导致数值解的不稳定，(ref 06 arm24) 中针对具体的应用做了详细解释。但在一些几何特征

(如分叉、凸起) 附近的奇异点又能有效地控制网格的扭曲，因此四边形网格应避免不必要的奇异点。

定义 1.1.2. *四边形网格的内部点中，度为 4 的称为规则点, 度不为 4 的称为奇异点。*

为了便于把四边形的质量表达成点坐标的函数进行优化，人们设计了很多代数度量。显然，这些度量需要是光滑的，具有全局的旋转、平移、伸缩不变性，并且当且仅当四边形为正方形时达到最大，四边形退化时达到最小。常用的代数度量都是基于网格单元与目标单元之间仿射变换的雅可比矩阵来定义的。

首先考虑任意 $\triangle v_1v_2v_3$，对应地，记它在网格中的坐标为 $\mathbf{x}_i, \mathbf{x}_i = (x_i, y_i)^T$, 它的目标单元的坐标为 $\mathbf{x}_i^g$, 它的参考单元坐标为 $\mathbf{x}_i^r$,$i = 1, 2, 3$，三者的变换关系见图。从参考单元到目标单元的变换见1.1, 参考单元到网格单元的变换见1.2, 目标单元到网格单元的变换见1.3。

$$\mathbf{x} = \mathbf{A}\mathbf{x}^r + \mathbf{x}_1 \tag{1.1}$$

$$\mathbf{x}^g = \mathbf{W}\mathbf{x}^r \tag{1.2}$$

$$\mathbf{x} = \mathbf{S}\mathbf{x}^g + \mathbf{x}_1 \tag{1.3}$$

容易验证，目标单元到网格单元仿射变换的雅可比矩阵，即 $\mathbf{S}$ 为

$$\mathbf{S} = \mathbf{A}\mathbf{W}^{-1} \tag{1.4}$$

其中

$$\mathbf{A} = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix} \tag{1.5}$$

$\mathbf{S}$ 与参考点的选取无关。对于以正方形为目标的四边形网格, 对应的按照1.1.1分成的三角形的目标为等腰直角三角形，此时目标单元即为参考单元，即 $\mathbf{W} = \mathbf{I}, \mathbf{S} = \mathbf{A}$。对于三角形网格，常把目标单元取为正三角形。

我们可以用雅可比矩阵 $\mathbf{S}$ 的条件数作为网格单元质量的度量。即

$$\kappa(\mathbf{x}) = \frac{\|\mathbf{S}\|_F \cdot \|\mathbf{S}^{-1}\|_F}{2} \tag{1.6}$$

其中 $\|\cdot\|_F$ 为矩阵的 Frobenious 范数，分母 2 为归一化因子，整个网格质量的度量为

$$|K_\kappa|_p(\mathbf{x}) = \left[\sum_{m=1}^{M} (\kappa_m)^p(\mathbf{x})\right]^{\frac{1}{p}} \tag{1.7}$$

1.1.3 2 维四边形网格生成的相关工作

由于全自动地生成高质量的四边形网格已经成为工业界的重大需求，近年来研究者们对这个问题提出了各种各样的解决方法。

基于三角网格的转化：由于高质量的三角网格很容易生成。一个最简单的想法是直接对每个三角形细分得到四边形网格，但这样产生了大量冗余的四边形单元，以及许多的奇异点，且网格质量较差，只可能被用在以四边形为主的网格中。另一类方法是合并两个有公共边的三角形为一个四边形，如 Johnston 等人 [1]Lo 和 Lee 等人 [2] 的工作。这类方法产生的四边形网格质量虽然高于前者，但仍然会产生较多的奇异点。总之，这里方法由于受到三角网格的点的分布和连接关系的限制，难以产生奇异点较少且高质量的四边形网格。

几何分解：这类方法先根据输入的 2 维区域的几何特征把它把分解为简单的、凸的区域, 再进一步把它们四边形化。Chae 和 Jeong【3】,Talbert 和 Parkinson[4],Nowottny[5] 使用了递归分解的算法。Baehmann 等人 [5] 使用了四叉树分解的算法。Tom 和 Amstrong[6],David[7] 等人利用中轴线结构分解了 2 维区域。这类算法能在产生较少奇异点的同时生成高质量的全四边形网格，但对于复杂的边界变化较为敏感，过多的细节使得算法不够鲁棒且难以自动化。

向前推进：这类方法由边界出发并向内推进，进而产生四边形网格。比如 blacker 和 stepheson【11】提出了 Paving 的方法，Kinney[12] 做出了改进。这类方法能对边界复杂的区域产生高质量的网格，但在不同方向推进的单元相遇时可能会发生自交，对网格的有效性没有保证，并且容易在边界附近产生较多的奇异点。

网格模板法：这类方法先产生特定的四边形网格模板，再把它应用于目标区域。如 Thacker 等人 [9] 第一个用格点自然形成的正方形网格结构作为目标区域内部的四边形，再对边界附近的正方形单元作出调整以适应不规则的边界。Haber[10] 等人忽略接近平角的拐角点，形成逻辑上 3、4 边的原始区域, 再利用无限映射 (transfinite mapping) 把正三角形、正方形上的四边形网格模板映到原始区域上。这类方法能在简单边界的区域产生高质量的网格，因此常常先用几何分解方法把复杂区域分解为简单区域，再利用合适的网格模板四边形化。

场：这类方法先在区域上产生定向场 (orientation field)，再由它指导产生分布接近这个场的四边形网格。它的优势在于场的奇异性直接决定四边形网格的奇异性，这使我们可以根据想要的四边形网格设计场。Fogg 等人【15】根据中轴角设计了定向场并由此产生少量孤立的奇异点，Kowalski 等人【16】先在产生符合边界几何特征的场，再通过解偏微分方程传播到内部，Sethian 等人 [16] 提出的快速行进 (fast marching) 算法可以把用户给定的稀疏场传播到整个区域。由场的奇异点出发的追踪 (tracing) 法可以分割得到简单区域，进而四边形化 (比如通过网格模板法)。

1.2 计算无翻转映射

映射和变形在几何处理、形状建模以及动画等领域中是至关重要的。我们首先给出

定义 1.2.1. 称映射 $f:\mathbb{R}^d \to \mathbb{R}^d$ 是**无翻转**的，是指映射的雅克比矩阵 $J(f)$ 的行列式 $det(J(f)) > 0$

在计算机图形学中，我们常常需要构造连续的分片线性映射，比如网格的变形、表面网格到平面的参数化、网格质量提高等等。这些问题要求映射得到的网格与输入网格具有相同的连接关系，并在无翻转的基础上最小化二者之间的某种几何度量的形变。由于翻转的单元无法通过物理变形得到，任何真实的材料无法被压缩为零甚至是负的体积或面积，所以无翻转的映射是一个基本而广泛的要求，尽管如此，保证这一点却很困难。

计算任意形体之间的无翻转映射 f 的一个常用方法如下：先设计一个能量方程 $E(f)$ 来度量对映射的某种需要，比如共形和面积形变，而 f 通常是点坐标 $\mathbf{x}$ 的函数，然后在无翻转和其它特定约束下 (如变形时用户指定的位置，网格质量提高时边界的固定) 计算最小化 $E(f)$ 的目标位置 $f(\mathbf{x})$，这是一个非线性带约束的优化问题。受到 (ref bv04) 的启发，许多工作通过设计一个能量函数，使其在 $J(f)$ 接近 0 时趋于无穷大，把问题转化为无约束优化，如 MIPS 能量 (Most-Isometric Parameterizations)(ref mips) 及其扩展 (ref extend mips)，并且致力于提出高效和收敛性好的优化算法。

在本文中有两处需要无翻转映射。一是把输入的 2 维三角网格变形为多正多边形结构的三角网格，这是对原始区域的简化。由于我们要利用变形的映射把多正多边形上生成的四边形网格映回原始区域，若映射含有较多翻转单元，最后会得到质量很差的四边形网格，我们采用了 (ref amips,xiaoming) 的能量防止翻转。另外一个需要无翻转映射的是四边形网格质量的提高，我们把待提高的网格看作由正方形网格变形得到的结果，然后固定边界，以最小化 $J(f)$ 的条件数为目标计算无翻转映射，即可得到不含翻转单元的高质量四边形网格。

1.3 本文的贡献及组织

基于1.1.2节和1.1.3中介绍的对 2 维四边形网格的要求以及相关的工作，我们提出了多正多边形辅助的 2 维四边形网格生成算法。

我们的算法是受到利用多立方体结构 (Polycube) 生成六面体的算法 (ref Gregson, PolyCut,Huang) 的启发，他们研究了自动构造多立方体结构的算法，计算出输入四面体网格到多立方体结构之间的映射，再利用多立方体结构自然的垂直特性，打上格点生成立方体网格，最后把它映回原始区域。这样得到的六面体网格内部完全没有奇异点，这限制了六面体网格的质量。本文的主要算法及写作组织如下：

(1) 第二章计算了由输入的 2 维三角网格到多正多边形结构之间的无翻转映射;

(2) 第三章利用基于中轴线的分割方法在多正多边形结构上生成四边形网格;

(3) 第五章借助计算的映射，把多正多边形结构上的四边形网格点映回原始区域，并保持网格的拓扑不变。

本文做出了如下的贡献:

(1) **提出了构造多正 N 边形结构三角网格的算法 ($N \geq 3$)** 输入不同的 N 以及高斯平滑的核宽度，可以不同程度地简化原始网格，进而控制四边形网格奇异点的多少。

(2) **完善了基于中轴线的四边形网格生成算法** 提出了把简单区域 (定义见3.2.2) 全部转化为 3、4、5-边界简单区域的算法，并结合 (ref m-a quad) 和 (ref integer) 实现了完整的四边形网格化算法。借助多正多边形结构的辅助，我们得到的中轴线对复杂边界不敏感，并且弥补了中轴线结构忽略几何特征的缺陷，对复杂的边界也能鲁棒地生成高质量、奇异点合适的四边形网格。理论上任何对边界敏感的算法 (如向前推进法和其他几何分割方法) 都可以通过多正多边形结构的辅助得到更好的结果。

(3) **提高四边形网格质量** 在 (sus2d) 的去除翻转单元同时提高网格质量的优化基础上，我们加上优化四边形角度的能量，通过增大它的比重可以得到正交性更好的接近长方形的单元。

第二章 多正多边形结构生成

输入一个 2 维三角形网格 **M**，本章的目标是把 **M** 变形为多正方形或多正八边形结构，并得到一个低形变的、三角形之间的一一映射。如果输入的是边界曲线，我们在边界上采点后使用 CGAL 库的共形 Delaunay 三角化。2.1节通过优化网格边界外法向的光滑和对齐的能量驱动网格变形，得到近似的多正方形和多正八边形结构。 2.2节通过给边界打标记得到边界上正确的拓扑结构，并以此为目标进行变形，最后压平边界得到最终的多正方形和多正八边形结构以及它和输入网格之间的映射。**??**节给出了总的变形算法和优化策略。对于边界法向稀疏的输入（比如整体上接近于矩形），把它变形为多正方形即可达到近似和简化的目的，但对于法向各异的边界，则需要变形为多正八边形，以保持基本的形态，避免过度地简化。

2.1 法向光滑和对齐的网格变形

指定一组目标法向，本节的目标是让输入网格边界的法向通过变形逐渐地与其最接近的目标法向对齐，并且使变形后的形状与输入尽可能地接近。我们采用法向对齐能量和刚性形变函数 [1] 驱动网格的变形 (2.1.1节)，并在变形的每一次迭代之前计算一个全局的旋转 (2.1.2)，使网格处在合适的坐标系中，尽量减小初始的法向对齐能量。我们用算法输入的核宽度 k 控制结果的奇异性。经过变形，与输入最接近的多正多边形结构会大致成型。

2.1.1 法向对齐能量

设输入的 2 维网格为 **M**，包含的三角形集合为 $\mathcal{T} = \{\mathbf{t}_1, \dots, \mathbf{t}_l\}$，顶点集合为 $\{\mathbf{v}_1^0, \dots, \mathbf{v}_m^0\}$，它的边界集合是 $\mathcal{B}$。$\mathcal{B}$ 包含 n 个三角形的边 $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$，它们的法向是 $\{\mathbf{n}_1, \dots, \mathbf{n}_n\}$。定义算子 $\mathbb{A}(\cdot)$，它将任意二维向量映射到离它最近的目标方向上。对于多正方形结构，目标方向为 $(\pm1, 0)^T, (0, \pm1)^T$ ，对于多正八边形结构，目标方向在原来的基础上多了 $(\pm\frac{\sqrt{2}}{2}, \pm\frac{\sqrt{2}}{2})^T$ 四个。于是多正方形或多正八边形定义为：$\mathbf{n}_i = \mathbb{A}(\mathbf{n}_i), \forall\, i$，即所有的边界法向都为目标法向的三角网格。在三维情形下，若目标方向为空间中的六个轴向，则生成的四面体网格为多立方体结构 (polyube)。因为 $\mathbb{A}(\mathbf{n}_i)$ 算子是作用在每条边上的局部算子，所以对于复杂的边界，直接对原法向作用此算子可能会在使边界被过度地分割，产生不必要的细节。因此我们先对法向做高斯平滑，再驱动网格变形。我们定义如下的法向对齐能量。

$$E_{align} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{b}_i \cdot \mathbb{A}(G_\sigma(\mathbf{n}_i)))2. \tag{2.1}$$

其中 $G_\sigma(\cdot)$ 为高斯函数：

$$G_\sigma(\mathbf{n}_i) = \sum_{\mathbf{b}_j \in \mathcal{K}_i} length(\mathbf{b}_j), \exp\left(-\frac{dist(\mathbf{c}_{\mathbf{b}_i}, \mathbf{c}_{\mathbf{b}_j})}{2\sigma_s^2}\right) \cdot \mathbf{n}_j. \tag{2.2}$$

其中 $\mathbf{c}_{\mathbf{b}_i}$ 是边 $\mathbf{b}_j$ 的中点，$dist(\mathbf{c}_{\mathbf{b}_i} - \mathbf{c}_{\mathbf{b}_j})$ 是从点 $\mathbf{c}_{\mathbf{b}_i}$ 到 $\mathbf{c}_{\mathbf{b}_j}$ 沿着边界的距离。设

$$\sigma_s = k\bar{l}. \tag{2.3}$$

其中 k 为算法输入的核宽度，$\bar{l}$ 为三角网格边界的平均长度。因此输入不同的 k，从而调整高斯函数的标准差，就可以控制结果的奇异性。

2.1.2 刚性形变能量

为了让变形后的多正方形或多正八边形结构近似输入的网格形状，我们考虑映射的刚性形变，使用 AMIPS 能量惩罚大的形变，并且避免变形过程中三角形的退化甚至翻转。设 $\mathbf{f} : R2- > R2$ 为一个分片线性函数，是输入网格到输出网格之间的映射，考虑任意三角形 $\mathbf{t} \in \mathbf{M}, \mathbf{f_t}(\mathbf{x}) = \mathbf{J_t x} + \mathbf{b_t}$ 是作用在 $\mathbf{t}$ 上的仿射变换, 其中 $\mathbf{J_t}$ 是 2 x 2 仿射矩阵，也是 $\mathbf{f_t}$ 的雅可比矩阵，设 σ_1, σ_2 为 $\mathbf{J_t}$ 的奇异值，则对于任意输入三角形 $\triangle \mathbf{v}_p^0 \mathbf{v}_q^0 \mathbf{v}_r^0$ 和对应的输出三角形 $\triangle \mathbf{v}_p \mathbf{v}_q \mathbf{v}_r$, 我们有

$$\mathbf{J_t} = [\mathbf{v}_p - \mathbf{v}_q, \mathbf{v}_p - \mathbf{v}_r] \cdot \left[\mathbf{v}_p^0 - \mathbf{v}_q^0, \mathbf{v}_p^0 - \mathbf{v}_r^0\right]^{-1}. \tag{2.4}$$

定义共形形变：

$$\delta_{conf,\mathbf{t}} = \frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1} = \frac{\mathrm{trace}(\mathbf{J_t J_t}^T)}{\det \mathbf{J_t}}. \tag{2.5}$$

定义面积形变：

$$\delta_{area,\mathbf{t}} = \left(\det \mathbf{J_t} + (\det \mathbf{J_t})^{-1}\right). \tag{2.6}$$

最后定义 *AMIPS* 能量：

$$E_{iso} = \sum_{i=1}^{l} \exp\left(\alpha\, \delta_{conf} + (1-\alpha)\, \delta_{vol}\right). \tag{2.7}$$

由于奇异值 σ_1, σ_2 表示，而雅可比矩阵的行列式 $\det \mathbf{J}$ 表示映射在面积上的缩放，故 (两个能量最小等价于奇异值为 1，行列式为 1)。另一方面，若任意一个三角形发生退化（行列式约等于 0）则上述 AMIPS 能量都会趋于正无穷。因此，AMIPS 能量能很好地表示映射的刚性形变的程度，并有效地惩罚较大的形变。若输入的三角形网格是高质量的，则 AMIPS 能量能保证产生无退化和翻转的三角形网格。

结合2.1和2.7，我们得到网格变形的总能量：

$$E_{deform} = E_{iso} + \lambda E_{align} \tag{2.8}$$

其中 λ 为权衡两个能量的权重，它的选取将在**??**节说明。E_{iso} 的变量为网格所有点的坐标。

2.1.3 降低法向对齐能量的预旋转

显然，如果网格的边界法向不沿着目标法向分布，法向对齐能量可能会很大。为了尽量减小法向对齐能量，我们希望先对网格进行全局地旋转，使得旋转后的网格边界的法向与它们的目标法向尽量接近。即：

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n} ((\mathbf{R}(\theta) \cdot \mathbf{b}_i) \cdot \mathbb{A}(\mathbf{n}_i))^2. \tag{2.9}$$

其中 θ 为旋转角，$\mathbf{R}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ 为旋转矩阵。

2.2 多正多边形结构的提取与网格压平

经过法向光滑和对齐的变形后，边界的法向已充分地和目标法向对齐，因此我们可以通过边界的目标法向提取出多正多边形结构。类似（gregson）,(2.2.1节) 采用两个策略对边界的目标法向做微小的修改，以得到正确合理的多正多边形结构。修改后再对网格进行变形，直到边界法向充分地与目标法向齐，最后 (**??**节) 对多正多边形结构的每条边加上严格的直线约束，压平网格，就可以得到最终的结构。

2.2.1 多正多边形结构的提取

为了描述多正多边形结构，我们首先定义三角形网格的段 (Segment) 和拐点 (Corner)。段是指多正多边形结构的边，拐点是指多正多边形结构的顶点。于是我们有：

定义 2.2.1. 三角形网格边界上彼此相连并且目标法向相同的一个边集称为一个段，记为 $\mathbf{S}$. 这个段的标记定义为它包含边的目标法向，记为 $\mathbf{l}$. 长度定义为边的个数, 记为 d. 所有段的集合记为 $\mathcal{S}$

定义 2.2.2. 两个相邻段的公共点称为一个拐点, 记为 $\mathbf{c}$。

这样，多正多边形结构网格定义为:

定义 2.2.3. 网格的所有边界边的法向等于其所在段的标记,即 $\forall i, \mathbf{S}_i \in \mathcal{S}, \forall j, \mathbf{b}_(ij) \in \mathbf{S}_i, \mathbf{n}_i = \mathbf{l}_i$

显然，多正多边形结构拓扑上的充分必要条件是相邻的段不能有相反的标记。另外，为了达到边界简化的目的，避免在四边形网格生成时产生过多的奇异点，我们不希望产生太短的段。于是我们要求：

(1)$\forall \mathbf{S}_i \in \mathcal{S}, l_i < k.$

(2)$\forall \mathbf{S}_i \in \mathcal{S}, \mathbf{l}_i \cdot \mathbf{l}_{i+1} \neq -1$

注 在不引起混淆的前提下，$\mathcal{S}_{i-1}, \mathcal{S}_{i+1}$ 表示 $\mathcal{S}_i$ 的两个相邻的段，$\mathbf{c}_i$ 所在的两个段为 $\mathcal{S}_{i-1}, \mathcal{S}_i$. 为了使多正多边形结构网格满足上述条件，我们采用如下策略修改边界的目标法向：

(1) 对长度小于 k 的段，用与它相邻的两个段通过泛洪填充 (flood filling) 的策略吞掉。即从段两端的边开始，依次向中间把边的目标法向修改为相邻段的标记，直到两条被修改的边相遇。

(2) 对相邻且标记相反的段，从二者公共的拐点开始在较长的段上截取一部分，形成一个新的段，以打断相反的标记。若较长段的长度 d 大于 $2k$, 则新段的长度取为 k，否则取为 $\lfloor \frac{d}{2} \rfloor$。通过新段的两个拐点形成向量 $\mathbf{v}$，从目标法向中选取不是原有的两个标记且与 $\mathbf{v}$ 的法向最接近的作为该新段的标记。

注意 执行策略 (1) 时，我们每次从所有段中选取最短的段；另外，为了不与策略 (2) 冲突，若与待吞掉的段相邻的两个段标记相反，则不对此段做处理。对于目标法向为奇数的多正多边形结构，不存在相反的法向，故不需要执行策略 (2)。

所有的段都满足上述两个要求后，固定边界的目标法向，以 (2.8) 为目标函数进行变形，其中 (2.1) 改为：

$$E_{align} = \frac{1}{n} \sum_{\mathbf{S}_i \in \mathcal{S}} \sum_{\mathbf{b}_j \in \mathbf{S}_i} (\mathbf{b}_j \cdot \mathbf{l}_i)^2. \tag{2.10}$$

当 $\mathbb{A}(n_j) = \mathbf{l}_i, \forall \mathbf{b}_j \in \mathbf{S}_i$ 时，再进行五次变形的迭代使网格充分接近正确合理的多正多边形结构，然后进入下面的网格压平。

2.2.2 网格压平

经过2.2.1节及变形后，每个段2.2.1上所有边的法向已经与该段的标记充分接近，因此只要强制每个段上的边界点都处在一条与该段的标记方向垂直的线段上，并让段的两个拐点成为线段的端点，就可以压平网格从而得到最终的多正多边形结构网格。为此，我们通过求解如下带线性约束的二次规划问题来确定拐点的坐标以及每个段对应的线段：

$$\begin{aligned} \min \sum_{i=1}^{m} \sum_{j \in \mathbf{S}_i} (y_j - l_i(x_j))^2 \\ s.t.\ \mathbf{c}_i = (x_i, y_i) = l_i \cap l_{i-1} \end{aligned} \tag{2.11}$$

其中 $\mathbf{c}_i$ 是段 $\mathbf{S}_i$□$\mathbf{S}_{i-1}$ 的公共拐点，$l_i(\cdot)$ 是段 $\mathbf{S}_i$ 所在直线，其方向由标记 $\mathbf{l}_i$ 唯一确定，j 遍历所有边界点。因此这个二次规划的变量为 m 条直线的截距、m 个拐点的坐标。

用拉格朗日乘子解得拐点坐标及段的位置后，我们直接更新当前拐点的坐标，并固定不是拐点的边界点的横坐标，带入其所在段的直线方程更新纵坐标即可。用拉格朗日乘子解得拐点坐标及段的位置后，我们直接更新当前拐点的

坐标，并固定不是拐点的边界点的横坐标，带入其所在段的直线方程更新纵坐标即可。为了解决压平可能带来的三角形的退化甚至翻转，得到输入和输出之间在三角形上的一一映射，压平后我们再固定网格的拐点，利用 (ref smooth and untangling) 优化其他点的位置。

2.3 网格变形算法和优化

本节给出从输入的三角网格到输出多正多边形结构网格的完整算法，并给出优化这些能量的策略。

极小化能量 E_{deform} 2.8之前，我们先极小化2.9，求出全局的旋转矩阵，然后作用在以减小初始的 E_{deform}.E_{deform} 是高度非线性的网格点坐标的函数，我们使用线回溯搜索步长的梯度法进行优化，并借助 HLBFGS 库 (ref hlbfgs) 实现，优化停止的条件是能量收敛或达到最大迭代次数。 2.8中 λ 表示 E_{align}(2.1) 所占比重, 因为初始网格的边界往往离目标法向差距较大，为了避免产生较大的形变或产生退化或翻转的三角形, 初始时我们让 E_{iso}(2.7) 占主导，取 λ 为 1，我们执行 5 次法向光滑和对齐的网格变形，每次使 λ 扩大 10 倍，这样就得到了边界和目标法向充分对齐的网格，然后按照2.2.1的方法提取正确的多正多边形结构，根据段2.2.1的标记，通过最小化2.8(其中 E_{align} 修改为2.10) 对不经过光滑的网格直接变形, 直到优化停止，此时边界法向和段的标记已经充分接近。最后我们借助 MOSEK 库 (ref mosek)，使用拉格朗日乘子法求解2.11, 从而压平网格, 再利用 (ref untangling) 提高网格质量。由此给出

input : triangle mesh M with vertices coordinates vector X,Gaussian kernel k,N
output: triangle mesh with regular polygon structure,M_p

$\lambda = 1$;
for $iter \leftarrow 1$ **to** 5 **do**
  $\theta^* \leftarrow \arg\min_\theta E_{rot}$;
  $X \leftarrow \mathbf{R}(\theta^*) \cdot X$;
  $\min_X E_{iso} + \lambda E_{Salign}$;
  $\lambda \leftarrow 10 \cdot \lambda$;
end
extract segments and corners;
swallow short segments;
if *N mod 2 is 0* **then**
  swallow opposite segments;
end
$\min_X E_{iso} + \lambda E_{align}$;
solve least square problem with linear constraints;
flatten mesh;
untangling and smooth;

算法 2.1: 多正多边形网格变形算法

第三章 基于中轴线的 2 维四边形网格生成

输入一个 2 维的边界不自交的有界区域，本章的目标是自动地在此区域上生成质量较高的四边形网格。若区域的边界是曲线，我们在边界上采点，用简单多边形逼近原边界。四边形网格生成分为两步：3.2节用 (ref 2d fem mesh generation) 的方法，基于区域的中轴线把它分割成充分简单、可直接产生四边形网格的子区域，3.3节先把子区域分割 3、4、5 边形结构，再利用 (ref integer programming) 整数规划的方法在每个子区域上产生四边形网格，并使得子区域网格之间彼此相容。

3.1 中轴线的概念

定义 3.1.1. *考虑 2 维区域 A 以及它的边界 ∂A, 对任意点 $P \in A$，记圆 $U_r(P)$ 是以 P 为圆心在 A 中的最大圆。所有 $U_r(P)$ 与 ∂A 交点至少为两个的点 P 构成的集合 $M(A)$ 称为的*中轴线*。其中 $U_r(P)$ 交于相同的两条边界的点 P 形成一条*中轴边*，中轴边的端点称为*中轴点*。*

中轴边和中轴点可被看成一个图的边和点，这个图就像 A 的骨架，代表了 A 的拓扑结构，因此可以作为区域分割的指导。

定义 3.1.2. *与 (ref fem2d) 等价地，我们把中轴点分为如下几类：(1)J 类：至少三条中轴边的交点;*

(2)I 类：两条中轴边的交点;

(3)C 类：中轴边与区域边界的交点;

(4)T 类：拓扑冗余，中轴边的终点，与边界无交点;

图中给出了各类中轴点的示例。(ref fem2d) 通过 Delaunay 三角化用三角形的外心连线逼近区域的中轴线，并根据三角形的邻接关系定义了与本文对应的几类三角形。本文利用 CGAL 库 (refCGAL) 精确地计算了中轴线。

3.2 基于中轴线的简单子区域的产生

本节复现了 (ref 2d fem mesh generation) 的方法，3.2.1先把区域分割为凸子区域的并，3.2.2再根据凸子区域的中轴线结构进一步把它们细分为充分简单区域的并。3.2.3对这种方法进行了讨论。

3.2.1 分割区域去除凹角

我们希望最终产生的每个四边形元素都接近矩形，且 2 维区域边界的拐角必然成为四边形网格的点，因此首先根据角度对它们进行分类，以在附近产生最

优的四边形分布。对每个角 θ, 提出如下的准则：

$$\begin{aligned} \min \sum_{i=1}^{n} (\theta_i - \frac{\pi}{2})^2 \\ s.t. \sum_{i=1}^{n} \theta_i = \theta \end{aligned} \tag{3.1}$$

即我们希望合理地分割 θ 使得附近的四边形尽量好，由这个准则可以把 θ 分为 3 类 (可使用表格，并画出示意图)：

(1)$\theta < 120^o, n = 0$, 度为 1;

(2)$120^o \leq \theta < 216^o, n = 1$, 度为 2;

(3)$216^o \leq \theta, n = 2$, 度为 3;

ref(2d fem) 称第 3 类角是凹的，即

定义 3.2.1. *角 θ 是凹的，是指 $\theta \geq 216^o$*

显然 (1) 的角可直接作为一个四边形的角，而 (2) 的角可被视为平角，让四边形直接通过附近的边界，3.3.2节将利用整数规划直接处理。但我们必须先合理地分割区域，使产生的子区域都只有 (1),(2) 类的角，根据3.1式，它们应尽量接近直角的整数倍。

(ref fem) 先在边界细致地采点，再以它们为约束对区域进行 Delaunay 三角化，然后从凹角点所在的内部边中选出最优的分割。选取的准则为

$$\min \sum_{i=1}^{4} (\theta_i - \frac{n_i \pi}{2})^2 + \mu |v - c|^2 \tag{3.2}$$

其中 n_i 是整数，θ_i 如图所示。与 (ref fem2d) 不同，我们加入了对另一个点 v 与其最近拐点 c 的距离的惩罚。因为这个点和附近拐点之间的边会成为至少一个四边形的边界，为了避免生成的网格过密，如果角度最优的分割点离另一个拐点很近，我们选择次优的角度，直接连接两个拐角点。我们不断选取边进行分割，并更新子区域的 Delaunay 三角化，直到所有的子区域都没有凹角为止。

3.2.2 基于中轴线的区域细分

给定一个所有拐点都不是凹角的 2 维区域，其中轴线就像该区域的”骨架”，并且很好地代表了区域的拓扑结构。根据**??**对中轴点的分类以及它们的连接关系，我们采取如下的一系列方法把这些子区域归结或进一步细分为简单区域 (ref def simple)。

(1)**T 类** 对 T 类中轴点附近区域的处理取决于沿着形成这段冗余拓扑结构的”圆弧”长成的角度总和，若它接近 π, 则插入一段分割分开这段”圆弧”和剩余的几何区域，一方面此”圆弧”和插入的分割形成了一个 2-边界简单区域，另一方面剩余的几何区域减少了一个 T 类中轴点，并增加了两个 C 类点。若整个

区域只有一个中轴点 T, 则这个区域被视为 1-边界简单区域。其他的情况 T 均可被视为**??**中第二类拐角点:

(2)**F 类** 若 C 类中轴点是 refsec:concavity removal 中第二类拐角点，我们重新称它为 F 点，任何与 F 点相连的 J 点被重新定义为 FJ 点，即由 $F-J$ 转化为 FJ，四边形网格将沿此处连续地通过 (图)。FJ 或 I 点意味着两条边被合并为一条逻辑上的边，因此在所有的操作中，我们忽略任何的 FJ 或 I 点，直接考察与它们相连的 C, F, J 点。

(3)**C 类** 若 C 点与 T 或 F 点相连，则此区域为 1-边界简单区域 (图)，若与 C 点相连，则此区域为 2-边界简单区域 (图)，若与 J 点相连，则我们之间考察：

(4)**J 类** 若 J 点与 F 点相连，则更改 J 为 FJ 点，若与三个 C 点相连，则此区域为 3-边界简单区域，若 J 点与两个 C 点相连，则被重新定义为 E 点，并忽略这两个 C 点 (图)。产生所有的 E 点后，我们考查与剩余 J 点相邻的 C, E 点个数：若有一个 C，两个 E, 则此区域为 5-边界简单区域，若有三个 E, 则此区域为 6-边界简单区域。若不满足如上所有情况，则此区域不能直接归为简单区域，将在 (6) 中处理。

(5)**E 类** 若与 E 相连的点为 C 点，则此区域为 3-边界简单区域，而 $E-E$ 结构的区域为 4-边界简单区域。

(6) **长链分割** 以上操作后若还有 $J-J$ 结构，则需要产生合适的分割，在分割两侧分别产生两个 E 点，分割后产生的两个子区域又可以归结为简单区域。若这条中轴线对应的边界有 F 点，则我们使用准则3.2再次从 Delaunay 三角形中选取分割，此时 F 点被当作凹点处理。若无边界点，则以中轴线上的点为圆心，沿着中轴线产生一系列内接圆，并以长度为极值的直径作为分割。

(7) **圆环** 最后我们要检查是否有逻辑上的圆环结构存在，若有，则我们类似 (6) 产生两条分割，即可产生两个 $E-E$ 结构，即 4-边界简单区域。

定义 3.2.2. *仅以 C 点为拐角点，根据逻辑上的边界数的不同，定义如下的简单区域 (图):*

3.3节将会先进一步处理简单区域，再用整数规划的方法得到四边形网格。

3.2.3 讨论与比较

(ref fem2d) 的方法先产生最优角度的非凹子区域，再根据子区域中轴线的结构进一步分解为简单区域进而四边形网格化。对任意输入的 2 维有界区域，它都能够生成四边形网格。

上节中，忽略 FJ 点是对中轴线的减支。3.3.1节将指出每个三支及以上中轴线的交汇点代表着至少一个奇异点的产生，因此对于有一定噪音的输入，忽略 FJ 点能有效地简化中轴线，避免生成不必要的奇异点。但对于抖动剧烈的噪音，此方法仍然会产生复杂的中轴线结构。本文的方法可以先把较复杂的输入简化为多正多边形结构，在它的多边形边界上做区域分割时，(1),(6)，(7) 类的情况不

会出现，奇异点的数目也会减少。通过改变多边形的边数，我们也可以控制奇异点的多少 (图：有效剪支前、后，我们的方法)

（ref fogg）指出，基于中轴线的方法存在的问题就是中轴线对几何特征不够敏感，图中三种差异较大的形状具有相同的中轴线结构，所以 (ref fem2d) 的方法最终都会产生质量较差的没有奇异点的四边形网格，但如果我们先把输入变形为多正八边形结构，三种形状的中轴线结构不同，因而会产生合适的奇异点，如果变形为多正方形结构，我们可以在同样没有奇异点的情况下产生质量更高的网格。

3.3 整数规划生成四边形网格

上节把 2 维任意有界区域细分为简单区域 (3.2.2)，本节先对简单区域中 0-边界、1-边界、2-边界，6-边界区域做合理地分割，全部归结为 3-边界、4-边界、5-边界简单区域，再利用 (ref integer) 提出的整数规划方法生成四边形网格。

3.3.1 合理的 3、4、5 边形区域的产生

与**??**节分割去除凹角的目标类似，我们要使新产生的角尽量接近直角, 并且希望尽量少地引入新的顶点和短边。因此，我们先对非 3、4、5-边界简单区域的边界采样做 Delaunay 三角化，再根据准则3.2从所有与 F 点相连的内部边中选出最优的一条作为分割。分割在两个两个新的区域中产生的 4 个中轴点直接记为 C 点。显然此时3.2中直角的整数倍 n_i 为 1。图 (split 0,1,2) 给出了分割示意图。

注 对于 6-边界简单区域，为了保证一定能分割为 3、4、5-边界区域，我们直接选取连接角度最大的 C 点和与它相对的两条边界上点的所有边，在其中选择最优的边作为分割，这样一定可以分解为图 (3+5,4+5,4+4) 中的一种。本文中采取的分割办法都是基于在 Delaunay 三角形中搜索最优边，但我们不需要为了找到最优分割过密地采点，因为**??**中通过优化的方法可以优化所有的非拐角点的位置来提高四边形网格的质量，而所有的分割以及整数规划中产生的新的点主要是为了决定四边形网格的拓扑结构，并使得优化有较好的初始。

3.3.2 整数规划生成四边形网格

给定 3、4、5-边界的简单区域以及它们对应的四边形网格的样式后，每条边被细分成四边形网格边的数目决定了产生的四边形网格。以简单区域的边的细分数为整数变量，对不同的网格样式加上特定的约束，再使相邻区域的公共边界相容，最后以最小化所求细分数与一组特定值 (比如用户给定的细分数，或基于之前的有限元分析的误差自动计算出的目标细分数 (ref integer 6)) 之间的差异为目标，就可以求得一组最优的细分数。给定符合网格样式的细分数后，基于映射的方法 (ref integer 3-5) 通过把简单区域映射为 3、4、5 边形并产生四边形网格，再映射回原区域，即可得到质量较高的四边形网格。本节给出了类似 (ref integer)

的目标函数和线性约束，不同的是，我们没有利用基于映射的方法决定内部点的位置，而是直接在简单区域上产生初始的四边形网格，再显式地以四边形网格质量为目标优化非拐角点的位置，得到最终的四边形网格。

变量 简单区域的边界上只有 C, F 两种点 (定义见**??**)，每两个相邻的点形成一条边，设 x 为它的细分数。每个 C 点和与它最近的 C 点形成一条简单区域的逻辑边，它由至少一条边形成，它的总细分数记为 N。

约束 3、5-边界简单区域生成的四边形网格分别产生度为 3、5 的奇异点，而 4-边界区域产生规则的四边形网格，没有奇异点。

3-边界区域的四边形网格样式如图所示，故需满足

$$\begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \tag{3.3}$$

其中 N_i, n_i 分别为外部和内部细分数。实际上，变量 n_i 是冗余的，因为上式等价于

$$\begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix} \tag{3.4}$$

又因为 n_i 是正整数，故上式等价于

$$\begin{cases} N_1 + N_2 > N_3 \\ N_2 + N_3 > N_1 \\ N_1 + N_3 > N_2 \\ N_1 + N_2 + N_3 = 2k(k \geq 1) \end{cases} \tag{3.5}$$

前三个式子保证了内部细分数为正，最后一个式子保证内部细分数为整数。

4-边界区域的约束很简单，只需保证相对的逻辑边细分数相等即可 (图)，即

$$\begin{cases} N_1 = N_3 \\ N_2 = N_4 \end{cases} \tag{3.6}$$

5-边界区域的约束与 3-边界类似，根据其四边形网格样式 (图)，我们有

$$\begin{pmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{pmatrix} \tag{3.7}$$

其中 N_i, n_i 分别为外部和内部细分数，同样地，上式等价于

$$\begin{cases} N_1 + N_2 + N_3 > N_4 + N_5 \\ N_2 + N_3 + N_4 > N_5 + N_1 \\ N_3 + N_4 + N_5 > N_1 + N_2 \\ N_4 + N_5 + N_1 > N_2 + N_3 \\ N_5 + N_1 + N_2 > N_3 + N_4 \\ N_1 + N_2 + N_3 + N_4 + N_5 = 2k (k \geq 1) \end{cases} \tag{3.8}$$

前 4 个式子保证了内部细分数为正，最后一个式子保证内部细分数为整数。

若只对每个简单区域加上独立的约束，则可能在公共边上产生不一致的结果 (图不相容-> 相容)。因此，相容约束要求每条公共边的细分数必须同时满足它所在两个区域的约束。由于我们在区域的细分过程中始终维持一个半边数据结构 (ref halfedge data structure, if any)，所以只要对每一组方向相反的半边设一个细分数变量，就可自动满足区域间的相容性。

目标函数和其他约束 我们希望产生质量较高且尽量稀疏的四边形网格，故对于每个简单区域，令其最短的边的目标细分数为 1，对其余边长与最短边长的比值向下取整，作为它们的目标细分数 (至少为 1)，然后规定解得的细分数不小于目标，并让它们和目标尽量接近。设 x_i 的目标细分数为 d_i, 则目标函数为

$$\min \sum_{i=1}^{n} x_i \tag{3.9}$$

再增加如下的约束

$$x_i \geq d_i \tag{3.10}$$

这样，以3.5,3.8,3.10为线性约束，**??**为线性目标函数，我们用 MOSEK 库 (ref mosek) 可以非常快的得到最优解。求出每条边的细分数后，我们直接在均分边界得到不是拐角点的边界点。对于 4-边界区域，直接连接对边上的对应点，并两组对边连线的交点作为内部点，即得到四边形网格。对于 3、5-边界简单区域，先利用边界上的 C 点求出它们的中心作为奇异点，再连接产生内部边，进而分别分割为 3 个和 5 个 4-边界简单区域，进而得到四边形网格。这些初始的四边形网格质量较差，且可能含有翻转单元，我们再利用5.2.2节的优化算法改变非拐角点的位置，得到有效且高质量的四边形网格。最后，我们可以根据需要通过把四边形单元一分为四得到更密的网格。

第四章　计算原始区域四边形网格

本章目标是借助点在三角形中的重心坐标，把多正多边形结构上生成的四边形网格映回原始区域。记原始三角网格为 $\mathbf{M}^0$, 其点集合为 $\mathbf{V}^0 = \{\mathbf{x}_1^0, ..., \mathbf{x}_n^0\}$，面集合为 $\mathbf{F}^0 = \{\mathbf{f}_1^0, ..., \mathbf{f}_m^0\}$，对应的多正多边形结构的三角网格为 $\mathbf{M}$, 其点集合为 $\mathbf{V} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$，面集合为 $\mathbf{F} = \{\mathbf{f}_1, ..., \mathbf{f}_m\}$，再 $\mathbf{M}$ 上生成的四边形网格点集合为 $\mathbf{Q} = \mathbf{q}_1, ...\mathbf{q}_l$。

由于任意的四边形网格点必落在 $\mathbf{M}$ 的某个三角形的边界或内部，我们可以用这个三角形的中心坐标表达这个点，即

$\forall \mathbf{q}_i \in \mathbf{Q}, \exists! \{\mathbf{x}_r, \mathbf{x}_s, \mathbf{x}_t\} = \mathbf{f}_j \in \mathbf{F}, s.t.$

$$\mathbf{q}_i = r\mathbf{x}_r + s\mathbf{x}_s + t\mathbf{x}_t \,, r + s + t = 1 \,, r, s, t \geq 0 \tag{4.1}$$

由于上式也是点落在三角形边界或内部的充分条件，故我们直接把重心坐标应用于原网格上的三角形，即计算出

$$\mathbf{q}_i^0 = r\mathbf{x}_r^0 + s\mathbf{x}_s^0 + t\mathbf{x}_t^0 \tag{4.2}$$

其中 $\mathbf{q}_i^0$ 是原始区域上的四边形网格的点，它也落在相应的三角形上，四边形网格的连接关系不变。由此可以看出，若第二章计算的映射形变较大，在原始区域上生成的四边形网格质量就会变差。最后我们可以利用第五章的方法提高网格质量。图给出了由重心坐标得到的四边形网格及优化后的结果。

第五章　2 维三角形或四边形网格的优化

固定网格的连接关系以及一些点位置的约束，网格优化技术通过移动自由点的位置最优化某种网格质量的度量。网格质量的度量与需求有关，(ref intro-meshquality) 定义了几种合理的度量。本章基于 (ref sus2d) 的方法，把得到无翻转网格和提高网格质量合并为一个优化过程。通过改变非拐角点的位置，就可以快速地得到无翻转且高质量的网格。

5.1　目标函数设计

设待优化的网格目标顶点的位置坐标记为 $\mathbf{x}$, 网格的边集为 $\mathcal{E} = \mathbf{e}_1, \mathbf{e}_2, ...\mathbf{e}_N$, 考查的三角形为 $\mathcal{T} = t_1, t_2, ...t_M$。对于四边形网格，我们先按 (ref quality) 所述把每个四边形分成四个三角形。为了提高网格质量，5.1.1节采用了 (ref sus2d) 中可以同时惩罚翻转单元和低质量单元的目标函数，5.1.2节设计了针对四边形网格的以直角为目标的网格度量。

5.1.1　同时解决翻转网格和提高网格质量的度量

我们在 (ref intro-meshquality) 中定义的度量虽然能表达一个有效网格的质量，但无法处理有翻转三角形的网格。这是因为它们无法连续地定义在整个 $\mathbb{R}^2$ 空间上，特别地，当某个三角形连续地从一个定向变为退化再变为另一个定向时 (ref intro-inverted)，这些度量函数的变化是不连续的。因此，许多网格优化的技术都需要为此定义不同的目标函数，先得到无翻转的网格，再提高网格质量，而这种方法无疑需要较多的迭代次数并且很容易收敛到较差的局部极小值。

考虑用雅可比矩阵 J 的条件数 κ(ref condional) 度量给定三角形与目标三角形的差距，首先将其重写为

$$\kappa = \frac{\|J\|_F \cdot \|J^{-1}\|_F}{2} = \frac{\|J\|_F^2}{2\sqrt{|\sigma|}} \tag{5.1}$$

其中 $\sigma = det(J)$。显然 κ 达到最大值 1 当且仅当这个三角形是等腰目标三角形，但 κ 在 $\sigma = 0$ 处不连续，即它不能表达退化或翻转单元的质量。

(ref sus2d) 对上述度量做了改进，上式中 σ 被替换为一个恒正且单调递增的函数

$$h(\sigma) = \frac{1}{2}(\sigma + \sqrt{\sigma^2 + 4\delta^2}) \tag{5.2}$$

因此有 $\delta = h(0)$。$h(\sigma)$ 和 σ 的函数图像见图。因此对三角形质量的度量更改为

$$\kappa^* = \frac{\|J\|_F^2}{2\sqrt{h(\sigma)}} \tag{5.3}$$

由此给出目标函数

$$E_{sus} = |K_\kappa^*|_p(\mathbf{x}) = \left[\sum_{m=1}^{M}(\kappa_m^*)^p(\mathbf{x})\right]^{\frac{1}{p}} \tag{5.4}$$

其中 κ_m^* 为改进的对三角形 t_m 的度量。

改进目标函数的合理性 虽然 κ^* 不是 J 的条件数，但由于 $\lim_{\delta\to 0} h(\sigma) = \sigma, \forall\delta \geq 0$ 且 $\lim_{\delta\to 0} h(\sigma) = 0, \forall\sigma \leq 0$，对于未翻转的网格单元，总可以选取充分小的 δ 使得 κ^* 与 κ 任意地接近。又由于 $h(\sigma)$ 关于 σ 在 $\mathbb{R}$ 上恒正且单调递增，所以目标函数可以对翻转单元做出更大的惩罚。另外，考虑到 $\forall\sigma > 0, \lim_{\delta\to 0} h'(\sigma) = 1$ 且 $\lim_{\delta\to 0} h^{(n)}(\sigma) = 0, \forall n \leq 2$, 容易证明修改后的目标函数的各阶导数与原函数有类似的收敛性，故我们可以使用基于一阶导数 (如梯度法) 和二阶导数 (如牛顿法) 的方法优化这个能量。

δ 的选取 在计算机实现中，δ 应取为在不引起任何数值计算问题的前提下的尽量小的正数。令 γ 是机器精度 $(0 < \gamma \ll 1)$, 为了避免计算目标函数5.4时计算机做除 0 运算，我们选取

$$\delta \geq \delta_{\min} = \begin{cases} \sqrt{\gamma(\gamma - \sigma_{min})} & if\ \sigma_{\min} < \gamma \\ 0 & if\ \sigma_{\min} \geq \gamma \end{cases} \tag{5.5}$$

其中 $\sigma_{min} = \min_{m=1,\ldots,M}(\sigma_m)$。由于 $h(\cdot)$ 的单调性，总有 $h(\sigma) \geq h(\sigma_{min}) \geq \gamma$，由上式也可以看出若网格无翻转或退化单元，则 δ 可取为 0，此时 $h(\sigma) = \sigma$，式5.4即为原始的目标函数 (ref quality)。

总之，通过选取小的 δ 使用改进的目标函数5.4，我们可以同时解决网格的翻转单元并提高网格的质量。

5.1.2 以直角为目标的四边形网格度量

(ref intro) 指出，四边形网格的角度往往被作为最直接的质量度量，因此在保证网格无翻转单元的基础上，我们再显式地加入

$$E_{ver}(\mathbf{x}) = \sum_{\theta}\cos^2(\theta) = \sum_{(\mathbf{e}_i,\mathbf{e}_j)}\left(\frac{\mathbf{e}_i\cdot\mathbf{e}_j}{\|\mathbf{e}_i\|\cdot\|\mathbf{e}_j\|}\right)^2 \tag{5.6}$$

其中 θ 遍历四边形网格的所有角，$(\mathbf{e}_i, \mathbf{e}_j)$ 遍历组成角的两条边的向量。对于无翻转单元的四边形网格，$0 < \theta < \pi, \forall\theta$。$\theta$ 越接近直角，E_{ver} 越小。此能量不能处理含有翻转单元的情况。

5.2 网格优化算法

本文有两处需要优化网格的质量。**??**节对2.2.2节对压平的多正多边形结构网格进行优化，5.2.2节对3.3.2节解整数规划问题后产生的初始四边形网格进行优化。

5.2.1 多正多边形结构三角网格的优化

由于2.2.2节压平网格时可能在边界处产生较差的网格，我们以正三角形为每个网格三角形的目标三角形,W 取为————————。并给出式5.3的梯度：

$$\partial_\alpha \kappa^* = 2\kappa^* \left[\frac{(\partial_\alpha J, J)}{\|J\|_F^2} - \frac{\partial_\alpha \sigma}{4\sqrt{\sigma^2 + 4\delta^2}} \right] \tag{5.7}$$

其中 $(\partial_\alpha J, J) = tr(\partial_\alpha J^T J), \alpha = x, y$ 为点的横、纵坐标。

优化策略 为了保持多正多边形的结构，我们固定拐角点的位置，并把边界非拐角点约束在边界上。我们使用梯度下降法优化能量 E_{sus}，每次局部地移动一个点，而固定其它点，使用线回溯搜索的方法确定步长，当前点的局部能量一旦下降，就转而优化下一个点。对于内部点，移动方向即为其梯度方向，对于边界非拐角点，把梯度方向在边界方向上的投影作为移动方向。算法终止的条件是所有点沿着目标方向的移动都不能使能量减小。

讨论 如果以所有点的坐标为变量，整体地优化 E_{sus}，则使能量下降的方向尤为重要。由于所有的点会同时移动，若任何点的子方向很差，则会导致整体的能量下降缓慢。若使用二阶的方法确定移动方向，则需要求解大规模的线性方程组，且会受到海森矩阵性态的影响，若使用全局的梯度作为近似的下降方向，会导致能量收敛性较差。局部地优化网格时我们不需要求解精确的下降方向，只需简单地用梯度方向近似并搜索出合适的步长。

5.2.2 四边形网格的优化

我们把网格的每个四边形分成四个三角形，然后以等腰直角三角形为目标三角形，此时 W 为单位阵。结合式5.4和5.6, 得到总的目标函数

$$E_{quad} = E_{sus} + \lambda E_{ver} \tag{5.8}$$

其中 λ 控制以直角为目标的能量所占比重。由于以等腰直角三角形为目标，取 $\lambda = 0$ 也能得到较好的角度。若初始网格含有翻转单元，我们先去掉 E_{ver}，采用上节所述的优化策略。一旦得到有效的网格，就加上 E_{ver} 继续优化，并在实行梯度下降时保证当前的网格无翻转单元。我们给出如下的算法：

参考文献

[1] Fu X M, Liu Y, Guo B. Computing locally injective mappings by advanced MIPS [J]. ACM Trans. Graph. (SIGGRAPH), 2015, 34(4):71:1–71:12.