

DOI: 10.7511/jslx20230920001

# 去除表面凸小特征的四面体网格质量优化算法

付营建<sup>1</sup>, 关振群<sup>1</sup>, 曹杰<sup>2</sup>, 赵国忠<sup>\*1</sup>

(1. 大连理工大学 工程力学系 工业装备结构分析优化与 CAE 软件全国重点实验室, 大连 116024;

2. 南华大学 数理学院, 衡阳 421001)

**摘要:**针对现有四面体网格优化算法难以解决表面凸小特征区域网格质量差的问题,提出了基于改进优化操作算子的去除表面凸小特征的四面体网格质量优化算法。首先利用边分裂对凸小特征区域网格进行插点,构建出凸小特征尺度方向的短边。然后利用节点合并对短边进行合并,并提出决定节点合并方向的减体积合并准则。针对合并过程中可能出现的错误四面体问题,给出了二分法调整节点位置的解决方法。节点合并后,形成新的网格拓扑,使网格不再覆盖凸小特征区域,从而达到去除表面凸小特征和提高网格质量的目的。最后,通过若干带有表面凸小特征的网格优化的典型算例,验证了本文算法的有效性。

**关键词:**小特征去除;四面体网格;网格质量优化;网格拓扑;边分裂

**中图分类号:** O302; O224

**文献标志码:** A

**文章编号:** 1007-4708(2025)01-0089-09

## 1 引言

近些年来,有限元方法在工程数值分析领域得到了非常广泛的应用,而有限元分析的前提是分析区域的网格生成,网格生成技术的快速发展为有限元方法在数值分析的应用提供了有力的保证<sup>[1-8]</sup>。

在网格生成过程中,模型表面小特征如凸小特征和凹小特征等区域容易生成质量较差的网格,从而影响整体网格质量。针对这个问题,目前大部分研究都是在几何层面上简化模型特征,以生成高质量的网格。Clark<sup>[9]</sup>提出了一种从 B-rep(boundary representation)模型中删除小特征的方法,利用底层实体建模内核提供的功能重建拓扑结构和几何形状,删除不需要的特征。曹秉万等<sup>[10]</sup>提出了一套实现组合曲面模型的自动特征简化算法,通过曲线和曲面合并实现小特征的去除。Kwon 等<sup>[11]</sup>利用体积分解操作将 B-rep 模型分解成多个体积块,然后根据其重要性依次去除分解的体积块,实现了模型的简化。

在实际应用中,很多模型都是通过逆向工程得到三维 CAD 离散数据模型,这种模型数据中缺少

几何信息。如果在几何层面上简化模型小特征,就需要利用离散模型重建几何信息,此过程可能会无法识别复杂或组合特征,造成主要特征的丢失<sup>[12]</sup>。而在网格层面上去除模型小特征,可直接对离散数据模型进行网格剖分,离散数据模型的主要特征可以得到完整保留。所以,在网格层面上去除小特征具有十分重要的工程意义。

四面体网格由于具有很强的几何适应能力<sup>[13]</sup>,在数值计算中得到大规模应用。对于工程中复杂几何模型,初始四面体网格生成质量经常难以达到求解器的要求,因此还需要进行网格优化。四面体网格优化算法大致可分为点光滑<sup>[14,15]</sup>、拓扑变换<sup>[16,17]</sup>和点插入/删除<sup>[18]</sup>三类,点删除也称为边收缩。现有的网格优化算法由于不会改变表面网格,在网格优化后,表面小特征区域的低质量单元依然存在。因此,对现有四面体网格优化算法进行改进十分必要。Mobley 等<sup>[19]</sup>提出了一种面向对象的几何特征抑制方法,并在有限元模型上成功进行了特征处理。Dey 等<sup>[20]</sup>提出了一种网格简化方法用于消除小特征对四面体网格模型的负面影响,该方法能确保修改后的网格相较于原始几何模

收稿日期:2023-09-20;修改稿收到日期:2023-11-09.

基金项目:国家重点研发计划(2023YFB3309100)资助项目.

作者简介:赵国忠\*(1972-),男,博士,教授(E-mail:zhaogz@dlut.edu.cn).

引用本文:付营建,关振群,曹杰,等.去除表面凸小特征的四面体网格质量优化算法[J].计算力学学报,2025,42(1):89-97.

FU Ying-jian, GUAN Zhen-qun, CAO Jie, et al. Tetrahedral mesh quality improvement algorithm for removing convex small features on surface[J]. Chinese Journal of Computational Mechanics, 2025, 42(1): 89-97.

型的有效性。现有网格层面上去除小特征的文献[19,20],都是利用现有优化操作算子去除小特征,并未结合小特征的具体特点,有针对性的对优化操作算子进行改进。

本文提出基于改进优化操作算子的去除表面凸小特征的四面体网格质量优化算法。首先,根据凸小特征区域网格的狭长特点,利用边分裂在网格的特定边插入特定位置的新节点,为凸小特征区域网格构建出凸小特征尺度方向的短边。然后,在允许对表面特征进行修改的前提下,利用节点合并对凸小特征区域网格的短边进行合并。其中,针对去除凸小特征问题的特点,提出决定合并方向的减体积合并准则;针对合并过程中可能出现的错误四面体的问题,给出二分法调整节点位置的解决方法。节点合并中通过构造新的网格拓扑,使网格不再覆盖凸小特征区域,从而去除凸小特征,同时凸小特征区域的低质量单元在节点合并后得以删除,网格质量得到提高。需要说明的是,本文算法的操作,如短边的构造方式和减体积合并准则等,主要是用来去除凸小特征的,对于其他类型的小特征问题,需要研究新的算法。最后本文通过若干带有表面凸小特征的网格优化的典型算例验证了本文算法的有效性。

## 2 边分裂

在进行节点合并前,需要利用边分裂为凸小特征区域的网格构建出凸小特征尺度方向的短边。边插点<sup>[18]</sup>是边分裂的基础,其属于点插入算法,在边插点中,新节点插入到单元的边上,网格中出现新边。边分裂在常规边插点算法的基础上做出了如下的改进,①对单元是否进行边分裂进行更精确的判定,减少对凸小特征区域以外的网格的修改。②调整四面体单元中要进行插点的边和插点位置,确保边分裂可以构建出凸小特征尺度方向的短边。

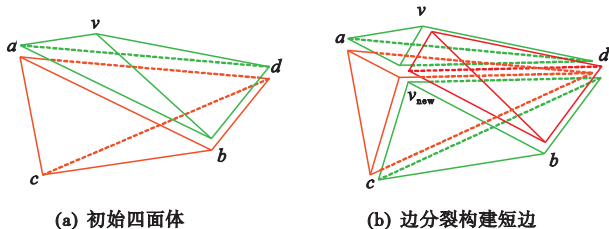


图1 边分裂  
Fig.1 Edge split

边分裂效果如图1所示。图1(a)中,四面体 $avbd$ 是表面凸小特征区域上的四面体单元,面

$abd$ 为凸小特征与模型的交界面,其法线方向为凸小特征尺度方向。图1(b)中,四面体 $avbd$ 经过边分裂,构建出了凸小特征尺度方向的短边 $vv_{new}$ ,单元 $avbd$ 也分为 $avv_{new}d$ 和 $bv_{new}vd$ 两个单元。

### 2.1 边分裂单元判定

传统的边插点算法以单元质量作为筛选条件,筛选出需要进行边插点优化的单元。由于采用单元质量作为筛选条件,会对较多凸小特征区域以外的单元进行边分裂。因此,根据凸小特征区域单元狭长的特点,边分裂以四面体单元最短垂线段长度作为筛选条件,并设置特征尺寸阈值 $l_{split}$ 。

单元的最短边长可能小于等于 $l_{split}$ ,对于这种单元,可以认为其最短边已经位于凸小特征尺度方向,不需要进行边分裂。如果单元的最短边长大于 $l_{split}$ ,则计算单元每个节点向其对边作垂线后得到的垂线段长度,如果最短垂线段长度小于等于 $l_{split}$ ,则对单元进行边分裂。

### 2.2 插点边和插点位置调整

传统的边插点算法尝试对每条边进行插点,但是对于凸小特征区域的网格,在任意边插点可能无法产生凸小特征尺度方向的短边。由于凸小特征区域的四面体单元较为狭长,且狭长方向垂直于凸小特征尺度方向,而单元的最短垂线段垂足所在边又与单元狭长方向相同。因此,边分裂选取四面体单元中最短垂线段垂足所在边进行插点,以构建出凸小特征尺度方向的短边。

如图2(a)所示,四面体 $avbd$ 是凸小特征区域上的单元,边 $ab$ 是 $avbd$ 中最短垂线段垂足所在的边,边分裂将在该边上插入新节点 $v_{new}$ 。在 $v_{new}$ 的位置选择上,边分裂不再选用图2(b)所示的传统中点,而是采用图2(c)所示的最短垂线段垂

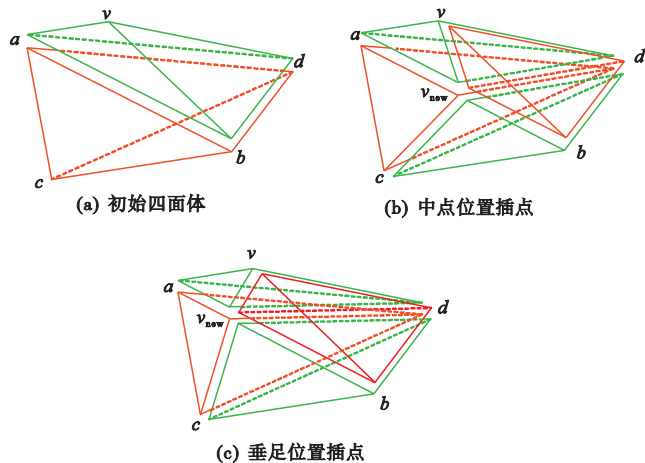


图2 边分裂插点位置  
Fig.2 Insertion point position of edge split

足。一方面,相较于边中点插点,垂足插点产生的新边  $vv_{new}$  更趋于凸小特征尺度方向(面  $abd$  的法线方向)。另一方面,垂足插点产生的新边  $vv_{new}$  长度更小,  $vv_{new}$  的节点在后续的节点合并中移动程度更小,节点合并具有更高的成功率。

### 3 节点合并

对于凸小特征区域上的网格的短边,对其两个节点(端点)进行节点合并,使网格不再覆盖凸小特征区域。边收缩算法是节点合并的基础,其收缩四面体单元的一条边,边的其中一个节点为保留节点,该节点在边收缩后得以保留,另一个节点在边收缩后删除。

在允许对表面特征进行修改的前提下,节点合并常在常规边收缩算法的基础上做出了如下改进,①制定减体积合并准则,为节点合并指定保留节点,确保可以通过节点合并使网格不再覆盖表面凸小特征区域。②利用二分法沿着合并边(两个端点要进行节点合并的边)调整保留节点的坐标,修正优化过程中产生的错误四面体。

节点合并效果如图 3(a)所示,四面体  $avv_{new}d$  和  $bv_{new}vd$  为覆盖凸小特征区域的网格,边  $vv_{new}$  的两个节点  $v$  和  $v_{new}$  将进行节点合并。图 3(b)中,经过节点合并,四面体  $avv_{new}d$  和  $bv_{new}vd$  删除,网格不再覆盖凸小特征区域。

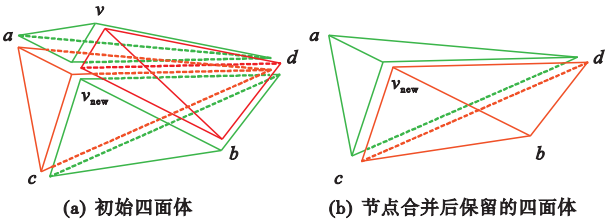


图3 节点合并  
Fig.3 Node merge

#### 3.1 减体积合并准则

图 4(a)中,狭长四面体  $avv_{new}d$  和  $bv_{new}vd$  为覆盖凸小特征区域的网格,节点  $v$  和  $v_{new}$  将进行节点合并,边  $vv_{new}$  为合并边。图 4(b)展示了节点合并中节点  $v$  得以保留后的四面体网格,图 4(c)展示了节点合并中节点  $v_{new}$  得以保留后的四面体网格。

传统的边收缩算法分别计算边的两个节点得以保留后网格的质量,网格质量更好的节点在边收缩后作为保留节点得以保留,即质量最佳作为边收缩准则。如果以质量最佳作为节点合并的合并准则,由于图 4(b)的网格质量要优于图 4(c)的网格

质量。因此,节点  $v$  在节点合并后得以保留,节点合并后的网格如图 4(b)所示。图 4(b)中,虽然两个狭长四面体删除,但是节点合并后的网格依然覆盖着凸小特征区域。

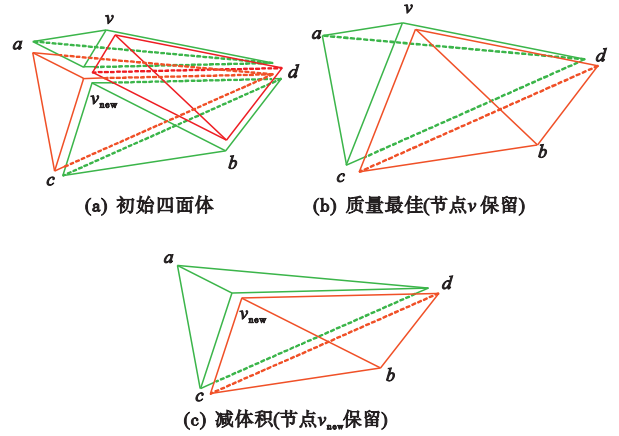


图4 节点合并方向  
Fig.4 Direction of node merge

为了确保节点合并后网格不再覆盖凸小特征区域,节点合并采用减体积合并准则:分别计算合并边的两个节点保留后,其相邻四面体的体积和,以体积和最小的节点作为节点合并中的保留节点。如果以减体积作为节点合并的合并准则,由于图 4(b)中节点  $v$  相邻四面体  $vcad$  和  $vcbd$  的体积和大于图 4(c)中节点  $v_{new}$  相邻四面体  $v_{new}cad$  和  $v_{new}bcd$  的体积和,因此,节点  $v_{new}$  在节点合并后得以保留,节点合并后的网格如图 4(c)所示,网格不再覆盖凸小特征区域。

同时,采用减体积合并准则后,图 4(a)中凸小特征区域的狭长四面体  $avv_{new}d$  和  $bv_{new}vd$  在节点合并后删除,图 4(c)的网格质量相较于图 4(a)也得到了提升。

#### 3.2 二分法调整保留节点坐标

传统的边收缩算法如果在收缩边的过程中出现错误四面体(负体积四面体<sup>[13]</sup>或体积为零的退化四面体),就会放弃对边的收缩。本文为了避免因错误四面体产生而放弃边收缩的情况,给出了通过二分法对保留节点的坐标进行调整的技术,修正优化过程中出现的错误四面体。

##### 3.2.1 错误四面体的产生

节点合并过程中,如果删除节点相邻的四面体和合并边相邻的四面体中出现了公共三角形面,则被删除节点相邻的四面体在合并后可能会成为错误四面体,而合并边相邻的四面体由于节点合并后体积一定为零,因此,这部分四面体从网格中直接删除。



如图5所示,边 $vv_{new}$ 为合并边,节点合并方向为节点 $v$ 合并到 $v_{new}$ ,节点 $v_{new}$ 为保留节点。四面体 $afvd$ 与被删除节点 $v$ 相邻,四面体 $avv_{new}d$ 与合并边 $vv_{new}$ 相邻。由于这两个四面体存在共面三角形 $avd$ ,被删除节点 $v$ 相邻的四面体 $afvd$ 在节点合并过程中可能会成为错误四面体。

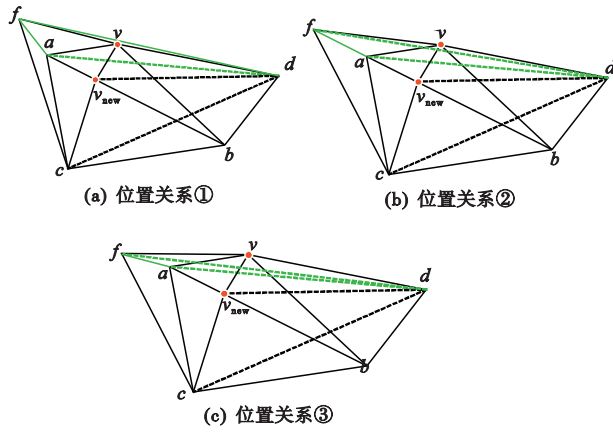


图5 三种位置关系

Fig. 5 Three kinds of positional relationship

四面体 $afvd$ 在节点合并过程中是否成为错误四面体,由节点 $v$ ,  $v_{new}$ 和三角形 $afd$ 所在平面的位置关系决定。节点 $v$ ,  $v_{new}$ 和三角形 $afd$ 所在平面的位置关系会出现如下可能:(1)节点 $v$ 与节点 $v_{new}$ 位于三角形 $afd$ 所在平面的同侧,定义为位置关系①,见图5(a);(2)节点 $v_{new}$ 位于三角形 $afd$ 所在平面,定义为位置关系②,见图5(b);(3)节点 $v$ 与节点 $v_{new}$ 位于三角形 $afd$ 所在平面的异侧,定义为位置关系③,见图5(c)。

对于位置关系①,节点合并后四面体 $afvd$ 不会成为错误四面体;对于位置关系②,节点合并后四面体 $afvd$ 会成为退化四面体;对于位置关系③,节点合并后四面体 $afvd$ 会成为负体积四面体。

### 3.2.2 保留节点的坐标调整

通过确定此类错误四面体的产生原因,可以发现,如果沿着合并边调整保留节点的坐标,可以将位置关系②和③调整为位置关系①,从而修正节点合并过程中出现的错误四面体。其中,保留节点坐标沿着合并边的调整可以采用二分法。

以图6(a)的位置关系②为例,节点 $v$ 和节点 $v_{new}$ 进行节点合并, $vv_{new}$ 为合并边, $v_{new}$ 为保留节点,利用二分法调整保留节点的坐标的迭代过程如下,将节点 $v_{new}$ 的坐标修改为边 $vv_{new}$ 的中点坐标,检查节点 $v$ 合并到 $v_{new}$ 后是否出现错误四面体,如果出现错误四面体,则恢复 $v_{new}$ 的原始坐标,并将 $v$ 的坐标修改为边 $vv_{new}$ 的中点坐标。重复此迭代

过程,直至合并过程中不再出现错误四面体。

图6为图5位置关系②和③中保留节点 $v_{new}$ 经过二分法调整后的效果。经过调整,位置关系②和③中的节点 $v$ 与节点 $v_{new}$ 位于三角形 $afd$ 所在平面的同侧,四面体 $afvd$ 在节点合并后不再成为错误四面体。

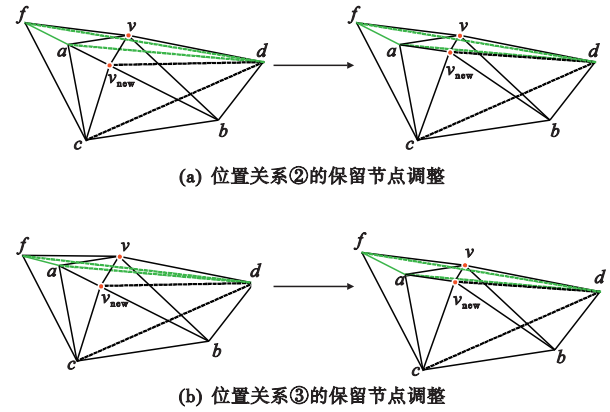


图6 二分法调整保留节点坐标

Fig. 6 Adjusting reserved node coordinates by dichotomy

### 3.2.3 小特征区域薄元的去除

原始网格拓扑和二分法调整保留节点坐标都可能导致节点合并后,小特征区域的网格表面产生薄元,这些薄元具有四个节点均位于表面的特点,为了减少这些薄元对网格质量的影响,在小特征区域将其去除。

考虑到薄元去除可能会改变网格表面形状,采用分层次方式去除薄元,尽量降低薄元去除对网格表面形状的影响。首先,利用拓扑变换算法<sup>[18]</sup>对小特征区域的表面单元进行优化,通过拓扑操作去除小特征区域大部分的薄元,图7展示了利用拓扑操作去除薄元。图7(a)的网格包含薄元 $abcd$ ,图7(b)展示拓扑操作修改后的网格,薄元 $abcd$ 删除,图7(a)中单元 $acbf$ 和 $adcf$ 经过拓扑操作变为图7(b)中单元 $adbf$ 和 $cbdf$ 。然后,对于拓扑变换算法处理后小特征区域依然存在的薄元,由于这部分薄元具有数量少、位于表面以及体积微小的特点,对表面形状的影响极小,因此,将其从网格表面删除,从而在尽可能降低对表面形状影响的前提下,将小特征区域的薄元完全去除。对于非小特征

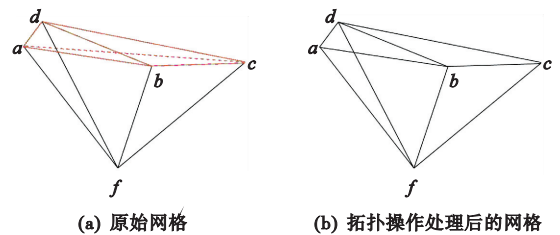


图7 拓扑操作删除薄元

Fig. 7 Deleting sliver elements by using topological operations

区域的薄元,由于不能对表面网格进行修改,理论上薄元无法完全删除<sup>[21]</sup>。

## 4 算法流程

### 4.1 节点合并算子

鉴于节点合并较为复杂,将其封装为独立算子,算子的输入为以节点号表示的合并边和初始四面体网格,算子的输出为节点合并后的四面体网格。

算子的具体流程如下。

Step 1. 根据减体积合并准则,判断合并边的节点合并方向,指定保留节点。同时,在此过程中预判断节点合并是否产生错误四面体,如果有错误四面体产生,执行 Step 2。

Step 2. 根据 Step 1 计算出的节点合并方向确定保留节点,利用二分法沿着合并边对保留节点的坐标不断进行调整,直至不出现错误四面体。

Step 3. 根据 Step 1 计算出的合并方向,利用边收缩算法在输入的四面体网格中对节点进行合并,输出节点合并后的四面体网格。

### 4.2 整体流程

结合边分裂和节点合并,本文算法整体流程描述如下。

Step 1. 找到网格中有大于两个节点位于表面的单元,如果单元存在一条边的长度小于等于特征尺寸阈值  $l_{split}$ ,或是存在一个节点向对边作垂线段,垂线段长度小于等于  $l_{split}$ ,则认为该单元为表面凸小特征区域上的单元,并将该单元记录到  $T_{feature}$ 。

Step 2. 对  $T_{feature}$  中每一个单元,找到其最短边,如果最短边长小于等于  $l_{split}$ ,则将该边记录到  $E_{merge}$ ,用于 Step 4 的节点合并。否则,记录单元到  $T_{split}$ ,  $T_{split}$  中的单元将进行边分裂。

Step 3. 对  $T_{split}$  中的每一个单元进行边分裂。找到单元中最短垂线段垂足所在边  $e$ ,利用边插点算法,以最短垂线段垂足为插点位置在边  $e$  上插入新点,并将产生的最短垂线段记录到  $E_{merge}$ 。

Step 4. 将  $E_{merge}$  中的每一条边作为合并边输入到节点合并算子中,对  $E_{merge}$  中的每一条边进行节点合并。

Step 5. 为保证完全清除凸小特征,循环执行 Step 1, Step 2, Step 3 和 Step 4,直到 Step 1 中计算出的  $T_{feature}$  为空。同时,为了提高算法效率,缩小 Step 1 中寻找小特征区域单元的范围。根据第

一次执行 Step 1 时计算出的  $T_{feature}$ ,计算出小特征区域的空间范围  $\Omega$ 。在后续循环执行的 Step 1 中,不再针对全体网格寻找小特征区域单元,而是在  $\Omega$  内寻找。

Step 6. Step 5 的循环优化结束后,对小特征区域的薄元进行去除。首先利用拓扑变换算法对空间范围  $\Omega$  内的表面单元进行优化,删除 Step 4 中产生的大部分薄元。然后,对于拓扑变换算法优化后小特征区域依然存在的薄元,因其对表面形状的影响极小,将其从网格表面中删除,从而在尽可能降低对表面形状影响前提下,实现小特征区域薄元的完全去除。

## 5 算例与分析

本文算法的核心优势在于对表面凸小特征的去除。本节首先针对不同的典型特征形状(圆形、三角形、长方形),测试了凸小特征去除效果,然后测试工程问题中复杂特征形状的凸小特征去除效果,同时对比优化前后的网格质量,验证算法的有效性。测试环境为 Inter Core i7-11800H (2.3 GHz)、32GB RAM、Microsoft Visual Studio 2017。

### 5.1 算例 1 典型特征形状的凸小特征去除

图 8 展示了圆形、三角形、长方形三种典型特征形状的凸小特征去除效果。图 8 左侧为初始网格,其中绿色单元所在区域为凸小特征区域。图 8 右侧显示了本文算法的去除效果,绿色线框为初始网格中凸小特征区域的轮廓线,可以看到三种典型特征形状的凸小特征均得以去除。表 1 展示了

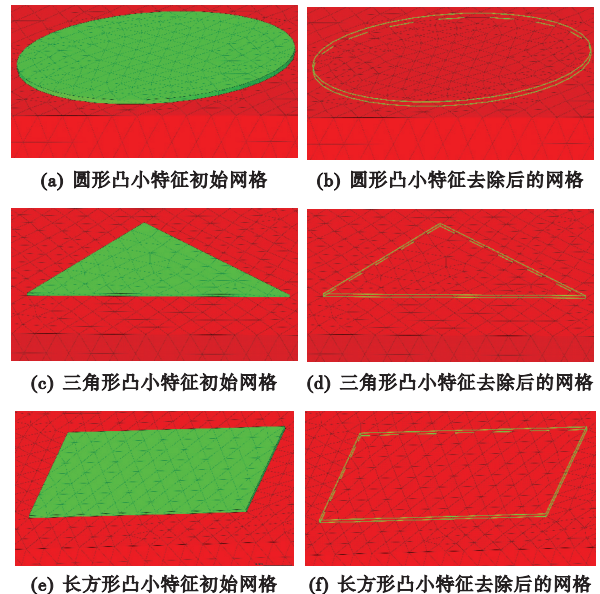


图 8 典型特征形状的凸小特征去除

Fig. 8 Convex small features removal for typical shape of feature

算例 1 的三个模型的整体单元数、优化前后小特征区域单元数以及本文算法的整体优化时间。

表 1 不同模型的单元数和优化时间

Tab. 1 Number of elements and optimization time of different models

模型	整体单元数	优化前小特征区域单元数	优化后小特征区域单元数	整体优化时间/ms
圆形特征模型	6085	840	0	53
三角形特征模型	6486	327	0	29
长方形特征模型	6721	600	0	46

## 5.2 算例 2 英文 LOGO 凸小特征的去除

如图 9(a)所示,算例 2 的凸小特征为某一品牌的英文 logo,其形状更为复杂。图 9(b)显示了本文算法流程 4.2 中 Step 1 至 Step 4 循环一次后的优化效果,绿色线框为凸小特征轮廓线。经过一次循环优化,表面的凸小特征基本去除,凸小特征区域仅有部分单元还存在。图 9(c)显示了 Step 1 至 Step 4 循环两次后的优化效果。



图 9 英文 LOGO 的去除过程

Fig. 9 Removal process of English logo

本算例的整体单元数为 62353,整体优化时间为 1353 ms,优化前小特征区域的单元数为 3381。第一次循环优化后,小特征区域的单元数为 58,优化时间为 1282 ms。第二次循环优化后,小特征区域的单元数为 0,优化时间为 47 ms。

## 5.3 算例 3 汽车模型的凸小特征去除

算例 3 为汽车模型,如图 10 所示,凸小特征分布在图中的 A, B, C 三个绿色区域。

由于三个凸小特征区域的形状基本相同,因此仅展示了凸小特征区域 A 的优化过程。图 11(a)为区域 A 的 112205 号节点附近的初始网格,相较于算例 1 和算例 2,算例 3 的凸小特征区域网格更为狭长。图 11(b, c)中的灰色单元由边分裂分成

了两个灰色单元,并成功构建出了凸小特征尺度方向的短边,其两个端点为 147573 号节点和 112205 号节点。图 11(c)中,短边的两个端点在节点合并处理下合并,112205 号节点保留,节点合并后的网格不再覆盖图 11(b)中两个灰色单元覆盖的凸小特征区域。图 11(d)为最终优化效果,网格不再覆盖凸小特征区域 A。

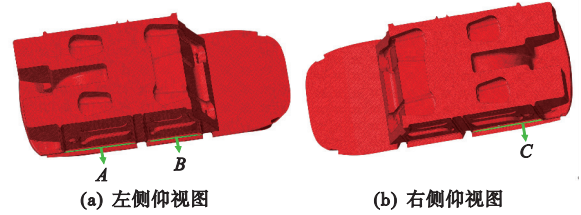


图 10 凸小特征区域分布

ig. 10 Regions distribution with convex small features

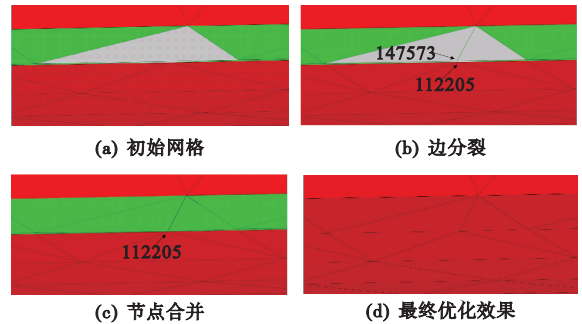


图 11 凸小特征区域 A 的优化

Fig. 11 Optimization of small feature region A

图 12 展示了 147645 号节点附近二分法调整保留节点坐标,修正错误四面体的效果。图 12(a)中,合并边的两个节点为 112287 号节点和 147645 号节点,保留节点为 147645 号节点。图 12(b)中,147645 号节点未经过二分法调整坐标,节点合并后,图 12(a)中的灰色四面体成为负体积四面体,灰色四面体的边 147645-112293 与红色四面体的边 112292-112267 出现交叉,节点合并后网格不相容。图 12(c)中,147645 号节点经过二分法调整,其坐标变为合并边的中点坐标,节点合并后图 12(b)中的错误四面体在图 12(c)中不再产生,节点合并后的网格依然相容。

本文算法流程 4.2 中 Step 5 的循环优化结束后,需要对小特征区域的薄元进行删除。本算例中,循环优化结束后小特征区域的薄元数约为 133,利用拓扑变换优化后,小特征区域的薄元数为 5,薄元数目减少了约 96%。对于剩余的 5 个薄元,由于其体积微小,对表面形状的影响极小,因此,将其从表面网格中删除,从而在尽可能降低对



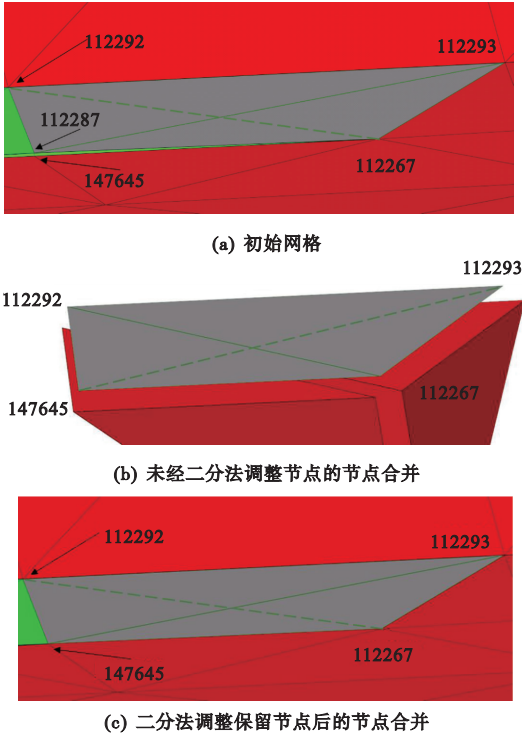


图 12 二分法调整节点修正错误四面体

Fig. 12 Adjusting node by dichotomy to correct the wrong tetrahedron

表面形状的影响的前提下,将小特征区域的薄元完全去除。图 13 展示了图 12(c)中薄元的去除,图 13(a)展示了去除薄元前的局部网格,薄元由 147645,112267,112293,112292 四个节点组成,图 13(b)展示去除薄元后的局部网格,经过拓扑变换优化,薄元删除。

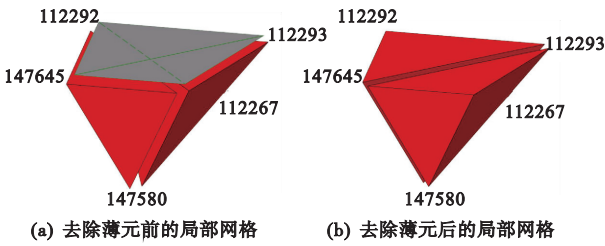


图 13 小特征区域薄元的去除

Fig. 13 Removal of the sliver element in small feature regions

本文算法在去除凸小特征的同时,自然会对网格质量有一个明显的提升,图 14 对比了优化前后网格的最小正弦值(minimum sine)<sup>[22]</sup>分布,最小正弦值取值范围为 $[0.0, 1.0]$ ,其值越大,网格质量越好。由于凸小特征区域的网格质量主要集中在 $[0.0, 0.03)$ 区间,因此图 14 对比了汽车模型在该质量区间的单元数目变化。优化前, $[0.0, 0.03)$ 区间的初始单元数目为 377,本文算法优化后,该区间单元数目为 4,数目减少了约 98.9%。同时, $[0.0, 0.006)$ 和 $[0.012, 0.0018)$ 两个区间的单元在本文算法优化后完全删除,网格的质量得到了明显提升。

本算例的整体单元数为 697782,小特征区域的单元数约为 440,优化后小特征区域的单元数为 0。汽车模型循环优化一次后,本文算法流程 4.2 的 Step 1 中计算出  $T_{feature}$  为空,结束循环优化。整体优化时间为 4018 ms,其中去除薄元的时间为 277 ms。

## 6 结论

本文提出了基于改进优化操作算子的去除表面凸小特征的四面体网格优化算法。在通过设置特征尺寸阈值筛选出小特征区域的网格后,利用边分裂和节点合并对小特征区域网格进行迭代优化,迭代流程如下,首先对小特征区域上的部分网格进行边分裂。在单元最短垂线段垂足所在边,以最短垂线段垂足为插点坐标插入新节点,构建出凸小特征尺度方向的短边。然后利用节点合并对网格中的短边进行合并。根据减体积合并准则计算合并方向,为节点合并指定保留节点。同时,在此过程中预判断节点合并过程中是否产生错误四面体。如果预判断有错误四面体产生,利用二分法沿着合并边不断调整保留节点的位置,直至不出现错误四面体。在确保节点合并不会产生错误四面体后,根据计算出的合并方向对短边的两个端点进行合并。最后重新筛选凸小特征区域的网格,如果筛选出的凸小特征区域网格数量为零,则停止迭代。对于节点合并过程中产生的位于小特征区域表面的薄元,在完成循环优化后,采用分层次方式去除小特征区域的薄元,尽量降低薄元去除对网格表面形状影响。首先,利用拓扑变换算法对小特征区域的表面单元进行优化,通过拓扑操作去除大部分的薄元,然后,对于拓扑变换算法处理后小特征区域依然存在的薄元,由于这些薄元对表面形状的影响极小,

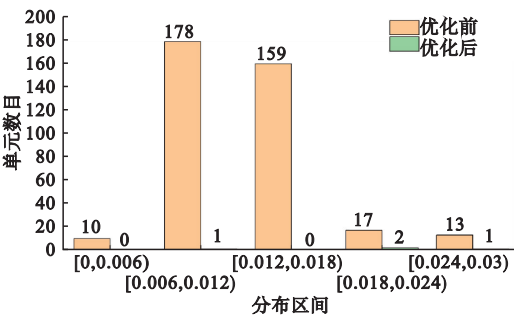


图 14 最小正弦值分布

Fig. 14 Minimum sine distribution

因此,将其从网格表面中删除,从而在尽可能降低对表面形状影响的前提下,实现小特征区域薄元的完全去除。

若干带有表面凸小特征的典型算例测试结果表明,①本文算法通过构造新的网格拓扑,使网格不再覆盖小特征区域,达到了去除典型和复杂特征形状凸小特征的目的。②本文算法在去除凸小特征的同时,能够删除凸小特征区域的低质量单元,提高网格质量。③在节点合并中,减体积合并准则能够指定合适的合并方向,使节点合并后的网格不再覆盖凸小特征区域;二分法沿着合并边调整保留节点位置能有效修正节点合并过程中产生的错误四面体。④本文算法可适用于复杂工程问题中的凸小特征去除。

## 参考文献(References):

- [1] 徐 权,冷珏琳,杨 洋,等.面向复杂几何模型的并行四面体网格生成方法[J]. 计算力学学报,2023,40(1): 140-145. (XU Quan, LENG Jue -lin, YANG Yang, et al. A parallel generation method of tetrahedral mesh for complex geometric models[J]. *Chinese Journal of Computational Mechanics*, 2023, 40(1): 140-145. (in Chinese))
- [2] 胡月凡,庞宇飞,肖素梅,等.一种新的准结构网格生成方法[J]. 计算力学学报,2019,36(2): 213-218. (HU Yue-fan, PANG Yu-fei, XIAO Su-mei, et al. A novel quasi-structural grid generation method [J]. *Chinese Journal of Computational Mechanics*, 2019, 36(2): 213-218. (in Chinese))
- [3] 王 硕,庞宇飞,肖素梅,等.基于网格框架的非结构附面层网格生成技术[J]. 计算力学学报,2021,38(6): 819-824. (WANG Shuo, PANG Yu-fei, XIAO Su-mei, et al. Grid generation technology of unstructured boundary layer based on grid frame[J]. *Chinese Journal of Computational Mechanics*, 2021, 38(6): 819-824. (in Chinese))
- [4] 曹 杰,赵铭伟,余 飞,等.高效可靠的边界层网格分块层进生成算法[J]. 计算机辅助设计与图形学学报,2020,32(8): 1199-1207. (CAO Jie, ZHAO Ming-wei, YU Fei, et al. Efficient and reliable advancing divided-layer method for boundary layer mesh [J]. *Journal of Computer-Aided Design & Computer Graphics*, 2020, 32(8): 1199-1207. (in Chinese))
- [5] 刘田田,董庆利,杨 洋,等.一种基于双前沿推进的边界层网格生成算法[J]. 计算力学学报,2022,39(1): 42-48. (LIU Tian-tian, DONG Qing-li, Yang Yang, et al. Boundary layer mesh generation method based on double-front advancing[J]. *Chinese Journal of Computational Mechanics*, 2022, 39(1): 42-48. (in Chinese))
- [6] Jiang M, Lu F S, Qi L, et al. A parametric mesh generation method for aero-engine compressor blades [A]. Proceedings of the 2019 Chinese Intelligent Systems Conference[C]. 2020.
- [7] 王 瑞,高曙明,吴海燕.六面体网格生成和优化研究进展[J]. 计算机辅助设计与图形学学报,2020,32(5): 693-708. (WANG Rui, GAO Shu-ming, WU Haiyan, Progress in hexahedral mesh generation and optimization[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2020, 32(5): 693-708. (in Chinese))
- [8] 徐亚飞,肖映雄,吴宇航.基于 Python-Abaqus 的自适应网格重划分算法实现及其应用[J]. 计算力学学报,2023,40(5): 807-814. (XU Ya-fei, XIAO Ying-xiong, WU Yu-hang, Implementation and its application of adaptive remeshing algorithm based on Python-Abaqus[J]. *Chinese Journal of Computational Mechanics*, 2023, 40(5): 807-814. (in Chinese))
- [9] Clark B W. Removing small features with real CAD operations[A]. Proceedings of the 16<sup>th</sup> International Meshing Roundtable[C]. 2008.
- [10] 曹秉万,赵大伟,陈建军,等.面向组合参数曲面扩展 B-rep 的特征简化和网格生成算法[J]. 计算机辅助设计与图形学学报,2014,26(1): 10-19. (CAO Bing-wan, ZHAO Da-wei, CHEN Jian-jun, et al. Defeaturing and meshing algorithms for extended B-rep of composite parametric surfaces[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2014, 26(1): 10-19. (in Chinese))
- [11] Kwon S, Mun D, Kim B C, et al. B-rep model simplification using selective and iterative volume decomposition to obtain finer multi-resolution models[J]. *Computer-Aided Design*, 2019, 112: 23-34.
- [12] 莫栋材.面向特征的网格模型简化算法研究[D]. 华中科技大学,2019. (MO Dong-cai, Research on feature-oriented CAD mesh model simplification algorithm[D]. Huazhong University of Science and Technology, 2019. (in Chinese))
- [13] 尚萌萌.多线程并行四面体网格优化算法[D]. 浙江大学,2017. (SHANG Meng-meng, Multi-thread Parallel Tetrahedral Mesh Optimization Algorithm[D]. Zhejiang University, 2017. (in Chinese))
- [14] 关振群,刘邦志,顾元宪,等.薄元分解与 Laplacian 光顺相结合四面体有限元网格优化方法[J]. 计算力学学报,2007,24(3): 257-263. (GUAN Zhen-qun, LIU Bang-zhi, GU Yuan-xian, et al. Tetrahedral mesh



- optimization method combining sliver decomposition and Laplacian smoothing [J]. *Chinese Journal of Computational Mechanics*, 2007, **24**(3): 257-263. (in Chinese))
- [15] Xi N, Sun Y J, Xiao L, et al. Designing parallel adaptive laplacian smoothing for improving tetrahedral mesh quality on the GPU[J]. *Applied Sciences-Basel*, 2021, **11**(12): 5543.
- [16] Joe B. Construction of three-dimensional improved-quality triangulations using local transformations[J]. *SIAM Journal on Scientific Computing*, 1995, **16**(6): 1292-1307.
- [17] Ma Y Z, Wang M N. An efficient method to improve the quality of tetrahedron mesh with MFRC[J]. *Scientific Reports*, 2021, **11**(1): 22802.
- [18] Klingner B M, Shewchuk J R. Aggressive tetrahedral mesh improvement[A]. Proceedings of the 16<sup>th</sup> International Meshing Roundtable[C], 2008.
- [19] Mobley A V, Carroll M P, Canann S. An object oriented approach to geometry defeaturing for finite element meshing[A]. Proceedings of the 7<sup>th</sup> International Meshing Round Table[C], 1998.
- [20] Dey S, Shephard M S, Georges M K. Elimination of the adverse effects of small model features by the local modification of automatically generated meshes [J]. *Engineering with Computers*, 1997, **13**(3): 134-152.
- [21] Chew L P. Guaranteed-quality Delaunay meshing in 3D[A]. Proceedings of the Thirteenth Annual Symposium on Computational Geometry[C], 1997.
- [22] Freitag L A, Ollivier-Gooch C. Tetrahedral mesh improvement using swapping and smoothing[J]. *International Journal for Numerical Methods in Engineering*, 1997, **40**(21): 3979-4002.

## Tetrahedral mesh quality improvement algorithm for removing convex small features on surface

FU Ying-jian<sup>1</sup>, GUAN Zhen-qun<sup>1</sup>, CAO Jie<sup>2</sup>, ZHAO Guo-zhong<sup>\*1</sup>

(1. State Key Laboratory of Structural Analysis, Optimization and CAE Software for Industrial Equipment, Department of Engineering Mechanics, Dalian University of Technology, Dalian 116024, China;

2. School of Mathematics and Physics, University of South China, Hengyang 421001, China)

**Abstract:** To solve the problem that it is difficult for existing tetrahedral mesh quality improvement algorithms to improve the quality of elements in convex small-feature regions on a surface, this paper proposes a tetrahedral mesh quality improvement algorithm based on improved optimization operators for removing convex small-features. Firstly, in order to construct short edges in the direction of the feature scale, new nodes are inserted into meshes in convex small feature regions by using edge split. Then, the short edges are merged by using node merge, and the reduced volume merging criterion is proposed to determine the direction of node merge. To avoid wrong tetrahedrons that may occur in the merging process, a solution to adjust the node position by dichotomy is given. After the node merge, a new mesh topology is formed so that the mesh no longer covers the convex small feature regions, which achieves the purpose of removing convex small features on a surface and improving the quality of meshes. Finally, several typical examples of mesh quality improvement with convex small features on a surface demonstrate that the proposed algorithm is effective.

**Key words:** small features removal; tetrahedral mesh; mesh quality improvement; mesh topology; edge split