

STL 实体的有限差分网格高效剖分算法

王跃平, 薛 祥

(哈尔滨工业大学 材料科学与工程学院, 150001 哈尔滨)

摘要: 为对形状复杂薄壁铸件进行精确、合理的网格剖分, 建立一种高效的 STL 实体有限差分网格剖分算法, 基于切片线扫描原理, 阐述了对多 STL 文件格式实体进行网格剖分的处理过程, 并编制了有限差分网格剖分程序. 程序采用动态数据结构、优化的奇异点处理技术以及先剖分后容错的方法进行三维有限差分网格剖分. 通过应用实例证明了该程序能够快速地对复杂形状 STL 实体装配体进行三维非均匀有限差分网格剖分. 该网格剖分算法具有计算速度快、变步长、奇异点优化处理和 STL 文件容错处理的特点, 剖分过程简洁迅速, 节省计算机资源, 剖分结果精确, 实用性强.

关键词: STL; 网格剖分; 有限差分; 数值模拟; 容错

中图分类号: TP391.9; TG2 **文献标志码:** A **文章编号:** 0367-6234(2012)05-0101-05

Effective algorithm for finite difference mesh generation based on STL solid models

WANG Yue-ping, XUE Xiang

(School of Materials Science and Engineering, Harbin Institute of Technology, 150001 Harbin, China)

Abstract: To obtain high quality finite difference mesh for castings with complex geometric shapes and thin walls, and to establish an effective algorithm for finite difference mesh generation of STL solid models, an enmeshment process of multiple STL models based on slice line scanning method was described and a computer program for finite difference mesh generation was developed. The program adopted dynamic data structure, updated technique of treating singular points and mesh filtering method for fault tolerance. It can be applied to generate 3D non-uniform meshes for assembled STL solid models rapidly. The algorithm is fast in running, alterable in mesh spacing, optimizational in singular points treating, and high fault tolerance. The mesh generation is rapid, easy to use, low memory needed, accurate and reliable.

Key words: STL; mesh generation; finite difference method; numerical simulation; fault tolerance

计算机数值模拟技术促进了铸造学科高速发展, 为提高和确保铸件质量开创了新局面. 目前, 国际上在铸造过程数值模拟领域都与网格剖分前处理技术密不可分^[1]. 精确合理的网格剖分是提高模拟精度、减少模拟时间的重要保障.

网格剖分模块可接受的输入数据一般为各种商品化造型平台均支持的标准格式, 其中 STL (Stereo Lithography) 格式^[2-3]应用最为广泛. 目前

比较成熟的有限差分网格剖分算法有: 切片法^[4-5]、射线穿透法^[6]、优化分层算法^[7]等. 本文在已有算法以及实现非均匀网格剖分、消除平行面误差等提高剖分精度的诸多研究^[8-9]基础上, 研究了铸造浇注系统有限差分网格高效精确剖分的算法, 编制了程序并给出了网格剖分实例.

1 基于 STL 的有限差分网格剖分实现

该算法基于切片线扫描^[10]原理进行有限差分非均匀网格剖分. 首先, 读取铸造过程数值模拟对象: 铸件、铸型、浇冒口、冷铁等各 STL 文件作相应分析, 对各实体各方向的平行面以及最小、最

收稿日期: 2011-06-15.

作者简介: 王跃平(1981—), 男, 博士生;

薛 祥(1961—), 男, 教授, 博士生导师.

通信作者: 王跃平, wpymlton@163.com.

大坐标值排序后作为实体的剖分段边界;然后,根据实体实际情况设置并自动调整剖分步长;最后,对各 STL 实体的剖分段进行网格剖分.

1.1 STL 文件读取分析

STL 文件格式将物体表面划分成很多空间小三角形,用这些三角形小平面来逼近原 CAD 实体,每个三角形用一个从实体内指向外部的单位矢量和 3 个顶点坐标表示. STL 文件有 ASCII 与 Binary 两种格式,其中 ASCII 格式 STL 文件中含有“facet”关键字,而二进制文件则不具有此关键字,所以“facet”关键字可以作为判断文件格式的依据,从而采用相应的读入接口程序. 本研究采用动态链表存储铸件、铸型、浇口等铸造模拟对象的 STL 数据为

```
struct STLData
{int flag[3]; //三角形面片属性标识
float facetnormal[3]; //三角形面片的法向量
float vertex[3][3]; //三角形的3个顶点坐标
struct STLData * next;
```

} * Dhead[NoSTLs]; //各 STL 数据的指针数组
对 STL 数据进行分析,用 flag 值作为小三角形面片属性标识: 如果 3 个顶点 Z 方向的坐标值均相等($z_{\max} = z_{\text{mid}} = z_{\min}$), 表明三角形为 Z 方向平行面, $\text{flag}[2] = 1$; 如果 $z_{\max} > z_{\text{mid}} > z_{\min}$, $\text{flag}[2] = 2$; 如果 $z_{\max} = z_{\text{mid}} > z_{\min}$, $\text{flag}[2] = 3$; 如果 $z_{\max} > z_{\text{mid}} = z_{\min}$, $\text{flag}[2] = 4$. 如果 3 个顶点 Y 方向坐标值均相等, 表明该三角形为 Y 方向平行面, $\text{flag}[1] = 1$; 如果 3 个顶点 X 方向坐标值均相等, 表明该三角形为 X 方向平行面, $\text{flag}[0] = 1$.

从模拟对象的 STLData 动态链表结构中提取实体的 X、Y、Z 平行面坐标以及最大、最小坐标 (X_{\min} 、 X_{\max} 、 Y_{\min} 、 Y_{\max} 、 Z_{\min} 、 Z_{\max}) 分别按照 X、Y、Z 从小到大排序并剔除重复点后存储在动态链表中为

```
struct STLParalelPlane
{float PPlane; //平行面坐标值
int No; //某方向剖分的网格起始编号
struct STLParalelPlane * next;
```

} * XPPhead [NoSTLs], * YPPhead [NoSTLs], * ZPPhead [NoSTLs], * AssemXPPhead, * AssemYPPhead, * AssemZPPhead; //各 STL 模型平行面数据(包括边界(Box[])处退化点或线的边界)的指针数组,以及铸造系统装配体平行面数据的指针

为了使所有的铸造模拟对象网格编号(i, j, k) 统一,把铸件、铸型等各 STL 实体的平行面数据分别按照 X、Y、Z 从小到大再次排序,剔除重复

点后存储在平行面链表 AssemXPPhead、AssemYPPhead、AssemZPPhead 中. 把 PPlane 作为剖分段的边界与网格单元的边界,能够有效避免平行面处误差,提高剖分精度;No 通过程序自动调整用户设置的网格剖分步长确定. 在处理平行面数据时,可能会遇到两个 PPlane 非常接近的情况. 经分析认为该问题是由于模拟对象的装配体文件转换成 STL 文件时的转换精度和算法精度不高造成的. 用户可根据需要设置一误差来解决. 例如设误差为 0.01 mm,若两平行面数据之差小于该值,则合并为一个平行面.

程序自动调整剖分步长的基本原理为: 设某剖分段的长度为 $L_i = (\text{PPlane})_{i+1} - (\text{PPlane})_i$, 用户设置的剖分步长为 dl , 如果 L_i/dl 不为整数, 即表明剖分段的边界与网格单元边界不重合, 需要对分区内的步长进行自动调整为

$$dl' = \frac{L_i}{f(L_i/dl + 0.49)}$$

式中: $f(x)$ 为取 x 的四舍五入的函数; dl' 为自动调整后的剖分步长. 某方向上网格参数设置、调整的示例如图 1 所示, 设用户设置的剖分步长均为 $dl = 5 \text{ mm}$.

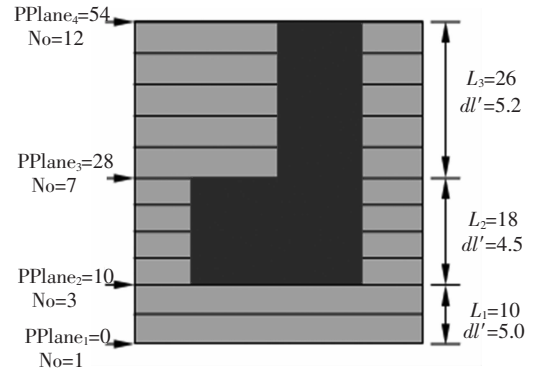


图1 某方向上网格参数示意图

网格剖分参数设置完成后,即可得各 STL 实体在 Z、Y、X 方向的起止网格编号 (N_{sz} 、 N_{ez} 、 N_{sy} 、 N_{ey} 、 N_{sx} 、 N_{ex}) 及起始坐标 (z_s 、 y_s 、 x_s), 然后按照 Z、Y、X 的顺序进行剖分.

1.2 Z 方向切割实体

Z 向切割平面 $z = z_k$ 切割实体, 获得一个或多个封闭轮廓线, 用 struct Section 存储这些线段为

```
struct Section
{double vertex0[2]; //线段第1个端点 x、y 坐标
double vertex1[2]; //线段第2个端点 x、y 坐标
double SignofNormaly; // 线段所在三角形面片法向量的 Y 向分量; Normaly
struct Section * next;};
```

切割从第 n 个 STL ($n = 1, 2, 3, \dots$) 的最小尺寸处 $z = z_k = z_s = ZPPhead[n] + dz[k] \times 0.5$ ($k = N_{sz}$) 开始,之后用 $z = z_k = z_s + (dz[k] + dz[k + 1]) \times 0.5$ ($k = N_{sz} + 1, N_{sz} + 2, \dots, N_{ez} - 1$) 平面切割实体的三角形面片. 在判断三角形面片与切割平面 $z = z_k$ 是否相交时,先选择非 Z 向平行面,即 struct STLData 中 flag[2] $\neq 1$ 的三角形面片,再比较三角形 3 顶点 Z 坐标值与 z_k 值:若 $z_{\min} \leq z_k \leq z_{\max}$,则该面片与 $z = z_k$ 切割面相交,否则不相交. 设三角形被切割棱边的顶点坐标为 (x_1, y_1, z_1) 、 (x_2, y_2, z_2) , $z = z_k$ 切割面与三角形棱边的交点坐标可以求得:

$$\begin{cases} x = \frac{z_k - z_2}{z_1 - z_2}(x_1 - x_2) + x_2, \\ y = \frac{z_k - z_2}{z_1 - z_2}(y_1 - y_2) + y_2. \end{cases} \quad (1)$$

把三角形面片与 Z 向切割面交线的两个端点数据存储在 vertex0、vertex1 中,并把当前三角形法向量的 Y 向分量赋给 SignofNormaly.

1.3 Y 轴方向切割封闭轮廓线

Y 向切割线 $y = y_j$ 切割 $z = z_k$ 面上封闭轮廓线,得到与一系列成对进出的扫描线段. 用 struct Scanbeam 记录这些线段端点的 x 坐标以及交点类型为

```
struct Scanbeam
{
    double vertexp; // Y 扫描线与封闭轮廓线的
    线段交点的 x 坐标
    int flag; // y 扫描线与封闭轮廓线的线段相
    交点类型
    struct Scanbeam * next; };
```

Y 向切割从第 n 个 STL 实体 ($n = 1, 2, 3, \dots$) 的最小尺寸处 $y = y_j = y_s = YPPhead[n] + dy[j] \times 0.5$ ($j = N_{sy}$) 开始,之后用 $y = y_j = y_s + (dzy[j] + dy[j + 1]) \times 0.5$ ($j = N_{sy} + 1, N_{sy} + 2, \dots, N_{ey} - 1$) 扫描线切割 $z = z_k$ 面上的封闭轮廓线,得到一系列交点,交点坐标可由式(1)求得. 设 struct Section 中记录线段的两个端点的 y 坐标分别为 y_{\min} 、 y_{\max} ,且 $y_{\min} \leq y_{\max}$,若 $y_{\min} \leq y_j \leq y_{\max}$,则该线段与 $y = y_j$ 扫描线相交,将交点类型赋给 flag:1 - 与含有 y_{\min} 的端点重合;0 - 交点在线段中间部分;-1 - 与含有 y_{\max} 的端点重合;2 - 与线段重合.

将所得交点按 x 坐标升序排序、处理奇异点、按奇偶配对,获得成对进出的扫描线段.

1.4 X 轴方向切割扫描线段

X 向切割扫描线段,得到 STL 实体在扫描线

段上的网格体心坐标 (x_i, y_j, z_k) . 若 $x = x_i$ 在成对进出的扫描线段上,则该网格为实体内部的网格. 根据多材质系统网格剖分时材质属性赋值方法^[11],为材质编号为默认值(空气,通常设为 0)的当前单元赋相应的材质编号值,保存网格信息,网格剖分基本完成. 算法流程如图 2 所示.

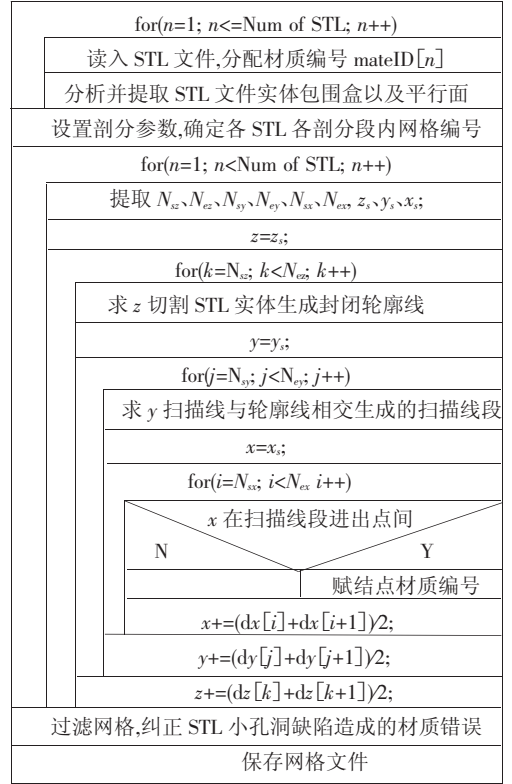


图 2 网格剖分流程图

2 网格剖分的关键点

2.1 动态数据结构的使用

网格剖分程序运行之前,并不能够确切地知道各 STL 实体的三角形面片数量、需要划分的剖分段数以及 XYZ 各方向的网格数等信息,这就给程序设计带来了一些不便. 在计算机内存资源一定的情况下,本程序采用动态链表和动态数组^[12]处理网格剖分相关数据,能够较大限度地提高网格剖分数量,增强了剖分程序的稳定性和实用性.

2.2 奇异点处理

奇异点是指切割面或扫描线与 STL 实体相交的特殊交点,可分为:1) Z 向切割 STL 实体时产生,如图 3(a)、(c)、(d)等 3 种情况;2) Y 向扫描线与封闭轮廓线相交时产生,如图 4(b)、(c)等 2 种情况.

Z 向切割奇异点处理为:图 3(a)中轮廓退化为点的情况,直接删除;图 3(c)中轮廓退化成直线或两点之间产生一条以上交线的情况,通过将切割面偏移一微小量消除; Z 向平行面作为剖分

段边界,可以排除情况如图 3(d). 通过上述处理,排除了封闭轮廓线退化成点或线的情况,避免了封闭轮廓线中出现重复线段的情况,保证了 Z 向切割面上的封闭轮廓线链表中记录的线段均唯一且 $l > 0$. 这样,程序更为简单,且能够有效地降低后续剖分的出错几率.

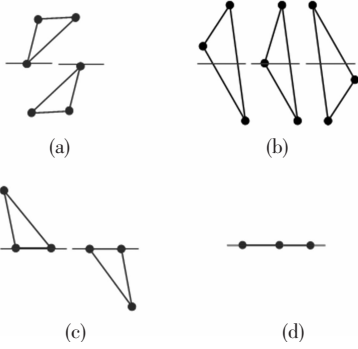


图 3 切割平面与 STL 三角形面片相交情况

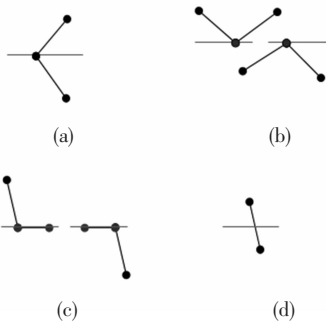
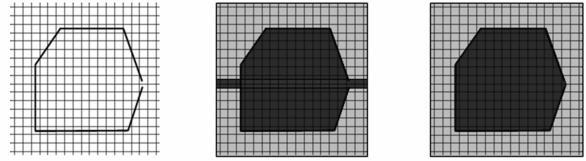


图 4 扫描线与封闭轮廓线交点情况

2.3 网格剖分对 STL 文件的容错处理

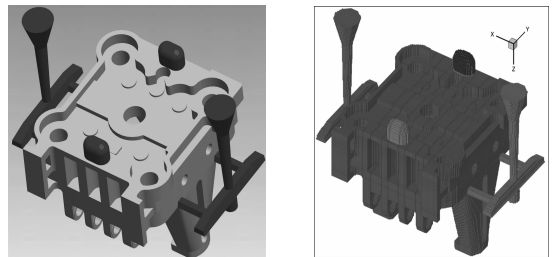
上述剖分算法原理简单,在 STL 文件完好的前提下十分可靠,但可能因转换精度以及其他算法精度的原因,STL 格式文件常包含各种细小的错误^[13],其中影响最大的是 STL 文件存在小孔洞. 因此要保证程序的稳定性,要么修复 STL,要么对这些错误进行容错处理. 本研究采用的是容错处理方法. 其基本原理为:若某条扫描线穿过封闭环上的小孔洞造成交点不配对的情况时,则给出警告:1)提示用户调整 Y 方向剖分步长,直到满足配对条件为止;2)保留体心坐标在该扫描线上的所有网格的原始材质(空气,材质编号为 0),待网格剖分完成后,再对网格进行过滤,纠正 STL 小孔洞造成的材质不匹配的网格单元. 其基本原理为:针对材质编号为初始值 0 的网格单元,搜索并判断其 3 个方向: X、Y、Z 上 6 个相邻网格单元的材质,如果某一方向上的两个相邻单元材质相同,则把该网格属性重置为相邻网格的材质编号. 原理示意图如图 5 所示.



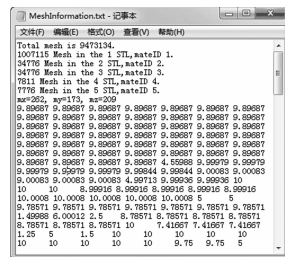
(a) 扫描线通过小孔洞 (b) 纠正前 (c) 纠正后
图 5 纠正 STL 小孔洞导致网格材质不匹配原理示意图

3 有限差分网格剖分实例

应用上述网格剖分算法编制了三维有限差分网格剖分程序. 该程序能够对各种复杂形状的 STL 实体进行网格剖分,剖分速度快、精度高、容错能力强. 图 6 为剖分实例,该铸造模拟对象包括铸件、浇冒口等多个 STL 实体,其中,铸件轮廓尺寸为 740 mm × 885 mm × 488 mm. 图 6(a)中的各实体主要由曲面组成,并且含有相对尺寸较小的倒圆、倒角、圆孔、凸台等. 若采用均匀网格,并且要保留尽可能多的几何特征,就必须采用较小的剖分步长. 这样网格数量会惊人地庞大,从而导致后续模拟计算时间显著增加. 本算法会自动在含有小几何特征的剖分段内采用较小的剖分步长,而其他剖分段内采用较大的剖分步长,在不显著增加网格数量的前提下进行剖分,且不会出现小特征丢失或网格不连通的情况. 网格剖分结果如图 6(b)所示,部分参数如图 6(c)所示. 设输入的最大空间步长为: $\Delta x = 10 \text{ mm}$, $\Delta y = 10 \text{ mm}$, $\Delta z = 9 \text{ mm}$,在 Pentium(R) 4 CPU 1.8 GHz,内存 512 MB 的普通计算机上剖分耗时 1.482 s,其中铸件网格数为 1 007 115 个.



(a) STL 实体装配体(铸件、浇冒口系统) (b) 非均匀有限差分网格模型



(c) 部分网格剖分参数

图 6 某大型复杂铸造模拟对象网格剖分实例

4 结 论

1) 该算法能对多 STL 模拟对象进行分区域非均匀剖分,程序能够自动调整剖分步长、有效消除平行面处误差,提高剖分精度。

2) 采用动态链表和动态数组处理数据、吸纳了现有算法处理奇异点的长处并进行改进,剖分原理简单,易于编程实现,在保证剖分精度的前提下,资源占用少,计算速度快。

3) 程序在网格剖分完成后,通过网格材质匹配对 STL 文件的进行容错处理,能够有效修复 STL 小孔洞造成的扫描线进出点不配对的错误,剖分结果精确、实用性强。

参考文献:

- [1] INOUE T, JU D Y. Analysis of solidification and viscoplastic stresses incorporating a moving boundary: an application to simulation of the casting process[J]. Journal of Thermal Stress, 1992, 15(1): 109 - 128.
- [2] ROSCOE L E. Sterolithography interface specification [M]. Valencia, CA: America - 3D systems Inc, 1988.
- [3] 侯华, 毛红奎, 张国伟. 铸造过程的计算机模拟 [M]. 北京: 国防工业出版社, 2008: 22 - 26.
- [4] 贾宝仟, 柳百成. 铸造数值模拟中直角六面体网格自动剖分技术[J]. 热加工工艺, 1996, (6): 27 -

29.

- [5] 邱宗文, 孙国雄, 刘永刚, 等. 基于 STL 的切片法有限差分网格剖分的研究[J]. 特种铸造有色合金, 2003, (6): 28 - 30.
- [6] 周建兴, 刘瑞祥, 陈立亮, 等. 基于 STL 的射线穿透法网格剖分的研究[J]. 铸造技术, 2001, (1): 15 - 17.
- [7] 徐宏, 钟雪友, 程军. 铸件三维有限差分网格自动剖分技术[J]. 铸造, 2000, 49(12): 903 - 907.
- [8] HUANG S H, ZHANG L C, HAN M. An effective error-tolerance slicing algorithm for STL files[J]. The International Journal of Advanced Manufacturing Technology, 2002, 20(5): 363 - 367.
- [9] 骆国建. 铸造过程计算机模拟前处理技术研究[D]. 武汉: 武汉理工大学, 2010.
- [10] 梁英业, 戴挺, 赵建新, 等. 基于 STL 的切片线扫描法网格剖分技术[J]. 铸造, 2005, 54(10): 1002 - 1005.
- [11] 关洋. 铸件充型凝固过程数值模拟前后处理技术研究[D]. 沈阳: 沈阳铸造研究所, 2004.
- [12] 陈凤翔, 李汪根. C + + 动态数组的实现与重用[J]. 计算机技术与发展, 2010, 20(2): 79 - 82.
- [13] SZILVÁSI-NAGY M, MÁTYÁSI G Y. Analysis of STL files[J]. Mathematical and Computer Modeling, 2003, 38(10): 945 - 960.

(编辑 张 红)

(上接第 100 页)

- [4] WEN Jyh-horng, LEE Gwo-Ruey, LIU Jia-wei, *et al.* Joint frame synchronization and channel estimation scheme using regression method in OFDM systems [C]//Proceedings of the Wireless Communications, Networking and Mobile Computing. Piscataway: IEEE, 2007: 164 - 167.
- [5] TANG S, PENG K, GONG K, *et al.* Robust frame synchronization for Chinese DTTB system[J]. IEEE Transactions on Broadcasting, 2008, 54, (1): 152 - 158.
- [6] WANG Jing, SHI Wenxiao, LI Haibo. Joint estimation of symbol timing and frequency offset for OFDM systems based on selective superimposed pilots [C]//Proceedings of the 8th International Conference on Electronic Measurement and Instruments. Piscataway: IEEE, 2007: 224 - 225.
- [7] SCHMIDL T M, COX D C. Robust frequency and timing synchronization for OFDM [J]. IEEE Trans Commun, 1997, 45(12): 1613 - 1621.
- [8] KISHORE C N, REDDY V U. A new method of frame

synchronization and frequency offset estimation in OFDM system [C]//International Conference on Signal Processing and communications. Piscataway: IEEE, 2004: 70 - 75.

- [9] CLASSEN F, MEYER H. Frequency synchronization algorithms for OFDM systems suitable for communication over frequency selective fading channels [C]//IEEE Vehicular Technology Conference. Piscataway: IEEE, 1994: 1655 - 1659.
- [10] LIU Hui, LI Guoqing. OFDM-based broadband wireless networks [M]. Xi'an: Xi'an Jiaotong University, 2008: 24 - 25.
- [11] MO R, CHEW Y H, TJHUNG T T, *et al.* New blind frequency offset estimator for OFDM systems over frequency selective fading channels [J]. Signal Process, 2007, 87(1): 148 - 161.

(编辑 张 宏)