DOI:10.16410/j.issn1000-8365.2024.4113

大型复杂钛合金铸件有限差分网格快速生成算法

石宇航[1,2],殷亚军[1],沈　旭[1],计效园[1],周建新[1]

(1. 华中科技大学 材料成形与模具技术全国重点实验室，湖北 武汉 430074；2. 华中科技大学 软件学院，湖北 武汉 430074)

摘　要：钛合金以其卓越的耐腐蚀性、轻质和高强度特性，成为航空航天工业中的核心材料。随着现代装备制造技术的飞速发展，钛合金铸件的制造技术也达到了新的高度，不仅复杂程度显著提升，而且铸件的规模也趋向于大型化。钛合金铸件大型化、薄壁化、复杂化的趋势，无疑对铸造过程的数值模拟带来了挑战。网格是数值模拟过程的计算域，因此首要问题就是如何高效生成高质量的网格。为应对这一问题，本文提出了一种算法，旨在高效生成适用于有限差分方法的六面体均匀网格。该算法结合了具有空间划分优势的八叉树数据结构和经典的射线法。首先利用八叉树数据结构对钛合金铸件的三维模型进行精细空间划分，通过与有限差分网格的映射关系，快速生成精确的表面网格。然后采用射线法，以表面网格为基础，依据射线与铸件内部的交点信息，精准构建内部网格。经过测试验证，该算法能够高效、准确地生成高质量的有限差分六面体均匀网格，为后续的数值计算提供了精确、可靠的计算域。

关键词：复杂铸件；钛合金；网格化；有限差分；八叉树

中图分类号：TG146.2+3　　**文献标识码**：A　　**文章编号**：1000-8365(2024)10-0917-08

Fast Finite-difference Mesh Generation Algorithm for Large and Complex Titanium Alloy Castings

SHI Yuhang[1,2], YIN Yajun[1], SHEN Xu[1], JI Xiaoyuan[1], ZHOU Jianxin[1]

(1. State Key Laboratory of Materials Processing and Die & Mould Technology, Huazhong University of Science and Technology, Wuhan 430074, China; 2. School of Software Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: Titanium alloys, with their excellent corrosion resistance, light weight and high strength, have become a core material in the aerospace industry. With the rapid development of modern equipment manufacturing technology, the manufacturing technology of titanium alloy castings has also reached a new height. Not only is the complexity significantly improved, but the scale of the castings also tends to increase. The trend of large-scale, thin-walled and complex titanium alloy castings undoubtedly brings challenges to the numerical simulation of the casting process. The mesh is the computational domain of the numerical simulation process, so the primary issue is how to efficiently generate high-quality meshes. To address this problem, this paper proposes an algorithm aimed at efficiently generating hexahedral uniform meshes suitable for finite difference methods. The algorithm combines the octree data structure, which has the advantage of spatial division, and the classical ray piercing algorithm. The octree data structure is first utilized to perform fine spatial delineation of the 3D model of titanium alloy casting, and an accurate surface mesh is quickly generated through a mapping relationship with a finite difference mesh. Then, the ray piercing algorithm is used to accurately construct the internal mesh based on the surface mesh and the intersection information between the rays and the interior of the casting. After testing and verification, the algorithm can efficiently and accurately generate high-quality finite difference hexahedral uniform meshes, which provides an accurate and reliable computational domain for subsequent numerical calculations.

Key words: complex castings; titanium alloy; mesh generation; finite difference; octree

收稿日期：2024-06-06

基金项目：国家重点研发计划(2022YFB3706800)

作者简介：石宇航，2001 年生，硕士生. 研究方向为铸造过程数值模拟前处理阶段的研究. Email: syhpuni@foxmail.com

通讯作者：殷亚军，1985 年生，博士，副教授. 研究方向为铸造过程数值模拟的研究. Email: yinyajun436@hust.edu.cn

引用格式：石宇航，殷亚军，沈旭，计效园，周建新. 大型复杂钛合金铸件有限差分网格快速生成算法[J]. 铸造技术，2024, 45(10): 917-924.

SHI Y H, YIN Y J, SHEN X, JI X Y, ZHOU J X. Fast finite-difference mesh generation algorithm for large and complex titanium alloy castings[J]. Foundry Technology, 2024, 45(10): 917-924.

中国知网　https://www.cnki.net

钛合金是一种强度高、密度小、耐腐蚀性好、耐热性高的合金材料[1],经过几十年的研究和开发,已被广泛应用于航空、航天、化工、冶金、能源等领域[2]。其中航空航天领域是钛合金应用占据主导地位的领域。钛合金是当代先进飞机和航空发动机的主要结构材料之一,可以大幅度提高结构减重效果和安全可靠性,是新一代飞机和新型发动机先进性的显著标志之一[3]。

钛合金由于自身的性质特点,导致使用机械加工、锻造、焊接等加工方法制造构件比较困难,特别是复杂的薄壁铸件[2]。目前钛合金构件主要采用铸造技术生产,其发展经历了硬模铸造、砂型铸造和熔模精密铸造 3 个阶段,其中熔模精密铸造工艺是为了满足航空航天领域对于复杂薄壁铸件的需要而发展起来的一种新型成型工艺[4],目前绝大部分航空领域的钛合金铸件都采用这种方法生产[5]。

铸造过程数值模拟技术在铸造领域发挥着重要作用。数值模拟技术可以预测温度场、流场及缩松、缩孔、卷气等缺陷,能够更好地理解和优化铸造充型和凝固过程,从而提高铸件的整体质量[6]。随着装备制造技术的发展,钛合金铸件趋向于大型化、复杂化和薄壁化,这给大型复杂铸件熔模铸造过程的精确数值模拟带来了挑战。数值模拟的过程包括前处理、数值计算和后处理 3 个步骤,其中前处理过程的作用是将空间上连续的计算区域划分成多个离散的计算区域[7]。数值计算的本质是将连续性介质上的偏微分方程问题转换为离散节点上的线性方程求解,网格作为数值计算的载体,其代表了计算量和计算区域[8]。前处理过程中生成的网格质量的好坏直接影响了数值计算过程。

经典的均匀网格生成算法使用的是射线法或者切片法。周建兴等[9]介绍了一种基于射线法的网格生成算法,使用一簇射线穿透模型,根据射线与模型的交点信息确定某个特定的位置是否位于模型内部,从而实现网格化。梁英业等[10]在文章中介绍了一种基于切片法的网格生成算法,切片法将模型沿着某个方面进行等距离切割,依据每一层的轮廓信息确定模型内部的位置,从而实现网格化。基于八叉树的网格生成算法已具备一些研究成果,例如吴晓军等[11]在文中提出,首先使用邻域搜索算法,对三角面顶点、三角面边和三角面进行体素化,然后使用 Flooding 算法对模型内部进行体素化。随着 GPU 的并行计算能力越来也强大,使用可编程图形硬件是另一种研究方向,Zhao 等[12]介绍了利用可编程硬件对三维模型表面体素化的思路,齐龙等[13]使用 CUDA 技术提出了一种 GPU 并行非结构网格生成技术。除了均匀网格生成算法,非均匀网格也是一个研究方向,戴愿桥[14]介绍了非均匀网格的发展过程,并且使用八叉树结构设计了一种自适应非均匀网格生成算法,该算法是在均匀网格划分的基础上,通过三角片面数量自动确定网格划分的停止条件,实现非均匀网格划分。

射线法和切片法作为传统的网格化算法,实现起来较为容易,但因涉及到大量的三角片面遍历操作,网格化过程较为缓慢。尽管使用 GPU 实现网格化算法可在很大程度上缩短网格化时间,但算法本身并没有改善,只是依靠 GPU 的强大算力实现了快速网格化,另外采用这种方式需要硬件支持。吴晓军等[11]提出的基于八叉树数据结构的网格化方法涉及到了编码、邻域搜索、对三角面顶点、三角面边、三角面、模型内部体素化,实现起来较为复杂。

本文介绍的网格生成算法将基于八叉树数据结构和射线法。该算法利用八叉树数据结构的特点,在将模型表面网格化的过程中将三角片面分别交给对应的八叉树叶子节点管理,从而极大减少了使用射线法进行内部体素化时需要遍历的三角片面的数量,实现了快速网格化。与传统的射线法和切片法相比,该方法极大减少了三角片面的遍历次数,在算法上实现了高效,同时又相较于吴晓军等[11]提出的基于八叉树的方法更加简单。另外,该方法也可以结合 GPU 等硬件实现更加快速的网格化。

1 有限差分网格生成算法

基于有限差分法的网格剖分通常是对 STL 格式的文件进行剖分[15],本文同样将以 STL 模型文件为基础,介绍一种针对有限差分方法的网格生成算法。该算法适用于大型复杂钛合金铸件,可快速生成高质量的六面体网格。在本算法中,网格生成的过程分为表面网格生成和内部网格生成两个部分。首先使用八叉树数据结构生成表面网格,再使用射线法生成内部网格从而实现网格化。

1.1 八叉树数据结构

八叉树数据结构是计算机图形学中广泛使用的一种三维空间的表示形式。八叉树的每一个节点表示一个正方体空间,每个节点有八个子节点[16]。八叉树采用递归的定义方式,通过持续对空间进行八等分实现对空间的表示和管理。八叉树结构如图 1 所示,其中最底层的根节点即为整个三维空间,叶子结点为分解结束时产生的最小空间。

文中使用的八叉树数据结构的伪代码定义

中国知网 https://www.cnki.net

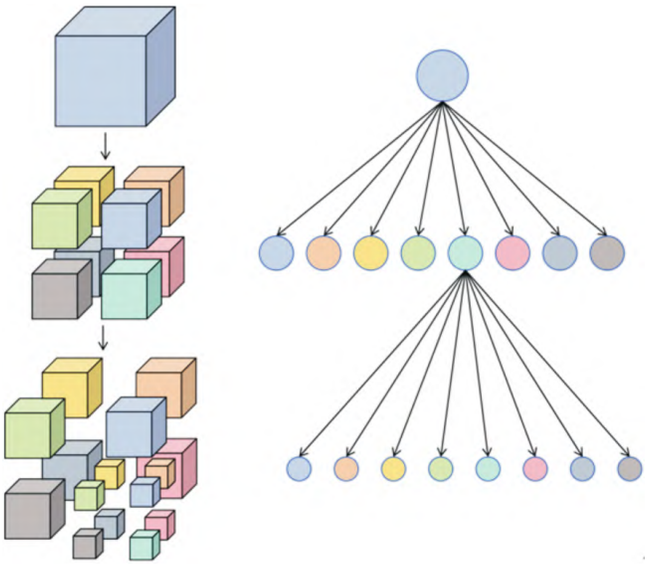


图 1　八叉树结构示意图

Fig.1　Octree structure

如下：

```
class Octree{
    Octree* children[8];
    vector<Triangle> triangles;
    double position[3];
    double length;
}
```

其中“children[8]”表示该八叉树结点拥有的 8 个子节点，“triangles”存储位于该节点空间中的三角形片面，“position[3]”表示该八叉树空间在三维空间中的位置，“length”表示八叉树空间的尺寸。

1.2　表面网格生成过程

有限差分六面体网格是三维空间被离散化后的结果，每 1 个离散的节点都是 1 个正六面体。八叉树数据结构的思想是将三维空间递归地划分为 8 个正六面体，这种性质可以用于有限差分网格生成。可将 1 个三维模型存放在 1 个八叉树的根节点中，将节点与模型的边界做碰撞检测，对产生碰撞的节点进行分解[17]，直至最高层节点的大小等于了网格大小。当分解结束后，最高层节点即构成了有限差分六面体网格。

STL 模型中保存了大量三角形片面的位置信息，这些三角形片面组成了三维模型的表面，即 STL 文件仅仅描述了模型的表面信息，因此使用八叉树结构对 STL 模型进行分解后获取的为表面网格。

表面网格的生成过程本质上就是递归创建八叉树数据结构的过程。当八叉树创建完毕后，叶子结点即为表面网格。表面网格生成流程如图 2 所示。

1.2.1　初始化数据结构

第 1 步，创建根节点，初始化子节点。初始状态下，认为 8 个字节点不存在，均为空值 NULL。

第 2 步，加载 STL 模型的三角片面，同时确定 STL 模型的边界值。读取 STL 模型文件，读取其中所有的三角形片面，存储至八叉树根节点的“trian

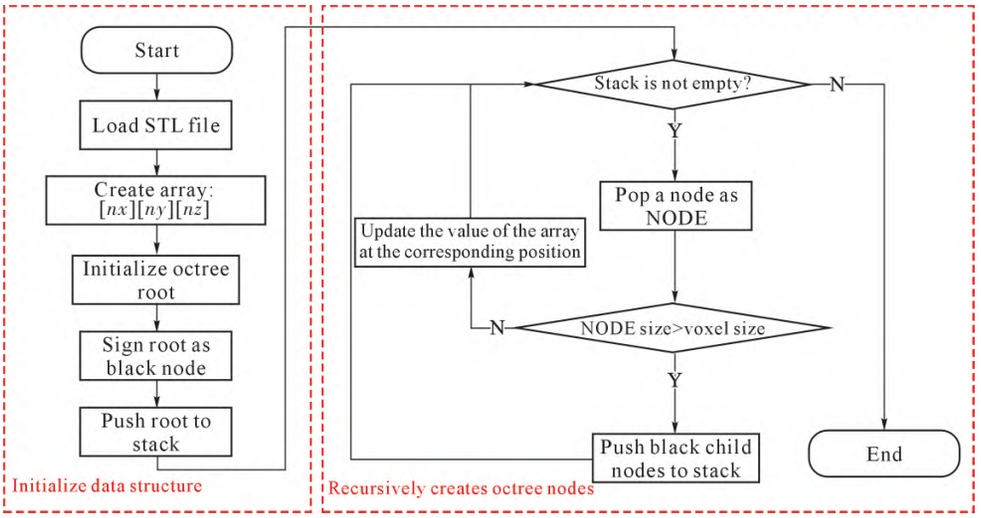


图 2　表面网格生成过程流程图

Fig.2　Flowchart of the surface mesh generation process

中国知网　https://www.cnki.net

gles”数组中，同时获取其中所有三角形的顶点坐标$[x, y, z]$，得到空间中最小坐标值$P_{min}=[x_{min}, y_{min}, z_{min}]$和最大坐标值$P_{max}=[x_{max}, y_{max}, z_{max}]$。

第 3 步，确定八叉树的位置和大小。依据P_{min}和P_{max}以及网格尺寸“size”确认八叉树根节点的位置“position”和八叉树根节点的大小“length”。八叉树根节点的空间必须能够包括整个 STL 模型，并且叶子结点的“length”应该等于网格大小“size”。因此“position”和“length”应通过式(1)~(8)所表示的条件计算。在本文中，令 position= $\{x_{min}, y_{min}, z_{min}\}$，令“length”为满足式(4)~(7)的最小值。

$$position[0] \leqslant x_{min} \tag{1}$$

$$position[1] \leqslant y_{min} \tag{2}$$

$$position[2] \leqslant z_{min} \tag{3}$$

$$position[0]+length \geqslant x_{min} \tag{4}$$

$$position[1]+length \geqslant y_{min} \tag{5}$$

$$position[2]+length \geqslant z_{min} \tag{6}$$

$$\exists n \in N_{+}, \frac{length}{2^n}=size \tag{7}$$

第 4 步，创建数组 voxel[nx][ny][nz]。三维数组 voxel 表示 1 个离散化后的三维空间，“0” 表示不存在网格，“1”表示存在网格，初始化时值全部置为 0。nx 表示 x 方向上网格的数量，ny 表示 y 方向上网格的数量，nz 表示 z 方向上网格的数量。nx、ny、nz 通过下式计算获得：

$$nx=\frac{x_{max}-x_{min}}{size} \tag{8}$$

$$ny=\frac{y_{max}-y_{min}}{size} \tag{9}$$

$$nz=\frac{z_{max}-z_{min}}{size} \tag{10}$$

1.2.2 递归创建八叉树结点

定义“白色节点”为不包含三角片面的字节点，“黑色结点”为包含三角片面的结点。在八叉树建立的过程中，如果三角片面位于节点内部，则该节点含有该三角片面，反之节点不包含该三角片面[18]。递归创建八叉树结点的过程就是找到黑色子节点的过程。递归停止的条件是找到所有满足“length=size”的黑色结点，这些黑色结点即为表面网格，将其信息保存在 voxel 数组中。递归的过程如下：

第 1 步，黑色子节点生成。当前的八叉树结点为 parent，遍历当前八叉树结点所存储的所有三角片面，对当前遍历到的三角片面与八个子节点所处的空间进行碰撞检测。如果某个子节点所处的空间与三角形片面产生碰撞，则创建该子节点并存储这些产生碰撞的三角片面至子结点的成员数组“triangles”中。三角片面与节点空间的碰撞检测方法有很多，例如林菲等[19]介绍了一种基于层次包围盒技术的碰撞检测方法，Akenine-Möllser[20]在他的文章中介绍了一种高效的碰撞检测方法。在本算法的实现过程中，采用该文献中介绍的实现方法。

第 2 步，递归处理 parent 拥有的所有黑色子节点。在第一步中，创建出的子节点均为黑色子节点。如果黑色子节点的 “length” 等于设定的网格尺寸“size”，则该节点不再继续分解，并将该结点在 voxel 数组中对应位置处设置为“1”。

递归过程的伪代码如表 1 所示。

表 1 递归创建八叉树子节点伪代码

Tab.1 Pseudocode for recursively creating sub nodes of an octree

No.		Pseudocode comment
1	function mesh(Octree parent)	▷Input octree root
2	if parent is NULL	▷If root is empty
3	return	▷Process end
4	for each triangle in parent.triangles	▷Traversing triangle facets
5	for each child in parent.children	▷Traverse the space of eight child nodes
6	calculate child.position and child.length	▷Calculate information of child nodes
7	if triangle collides with the space child	▷Triangle collides with facets
8	if child is null	▷If child node is not created
9	create child	▷Create the child node
10	add triangle to child.triangles	▷Add the triangle to triangles of the child node
11	if child.length == voxel size	▷Recursive termination condition
12	calculate child's index {x, y, z} in the voxel[nx][ny][nz]	▷Map the node to voxel array
13	set voxel[x][y][z] = 1	▷Save voxel information
14	return	▷Process end
15	else	
16	for each child in parent.children	▷Traverse child nodes
17	mesh(child)	▷Recursive processing of child nodes

中国知网 https://www.cnki.net

1.3 内部网格生成过程

在表面网格生成执行结束后，得到了 1 个三维数组 voxel[*nx*][*ny*][*nz*]和 1 个八叉树空间结构。voxel 中值为 1 的位置与八叉树中的黑色结点是一一对应的，通过 voxel 数组的一组{*x*, *y*, *z*}值可以找到八叉树中对应的黑色叶子节点。

内部网格在已经获取的表面网格的基础上，使用射线法的思想生成。射线法的原理和实现在文献[5]中已介绍过，文中沿 *z* 轴方向发出射线，合计发射出*nx***ny* 条射线，从而穿过 voxel 数组的表示的所有离散空间。射线发出后，记录射线与三角片面发生碰撞的坐标位置。依据记录下的碰撞坐标值生成内部网格。内部网格生成过程的流程图如图 3 所示。

内部网格生成的伪代码如表 2 所示。

内部网格生成完毕后，voxel 数组中保存的即为完整的网格信息，其中“0”表示没有网格，“1”表示存在网格。

2 网格化结果与分析

使用 C++ 语言实现上述算法编写单线程程序，并使用该程序将大型复杂钛合金铸件模型及其浇注系统进行网格化，使用 AMD Ryzen 5 5600 6-Core 处理器执行该程序。使用开源三维图像处理库 Visualization Toolkit 显示模型与网格，对网格化结果进行分析。

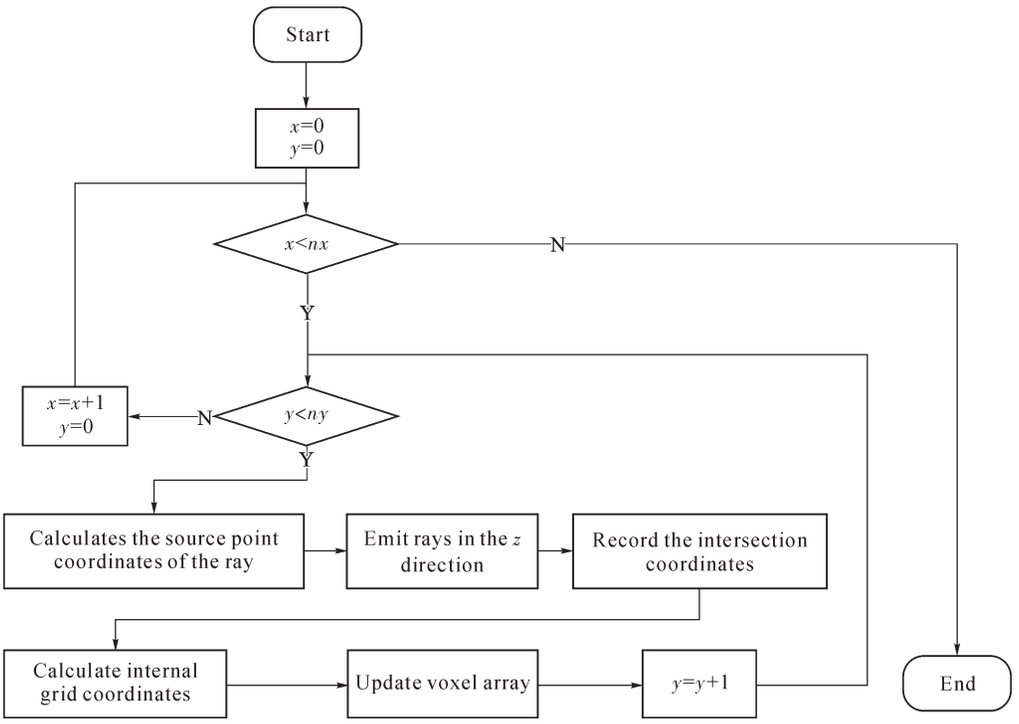


图 3 内部网格生成流程图

Fig.3 Internal mesh generation flowchart

表 2 内部网格生成伪代码

Tab.2 Internal grid generation pseudocode

No.	Pseudocode comment
1 function padding()	
2 for each x in range(0, nx)	▷Traverse along the x-axis direction
3 for each y in range(0, ny)	▷Traverse along the y-axis direction
4 calculateRayOrigin(x, y)	▷Calculate the source point of the ray
6 for each z in range(0, nz)	▷Traverse along the z-axis direction
7 if voxel[x][y][z] is black	▷If it is a black node
8 leafNode = getLeafNode(x, y, z)	▷Get triangular facets
9 for each triangle in leafNode.triangles	▷Å Traversing Triangle Faces
10 if ray intersects triangle	▷If ray intersects triangle
11 store(intersectionPoint)	▷Storage intersection points and remove duplicate
13 calculateInnerVoxelPositions(intersectionPoints)	▷Calculate internal voxel coordinates
14 updateVoxelArray(innerGridPositions)	▷Update the voxel array

中国知网 https://www.cnki.net

2.1 模型介绍

采用钛合金机匣模型测试该程序。钛合金机匣模型如图 4 所示,包含 291 838 个三角形片面,875 514 个顶点。

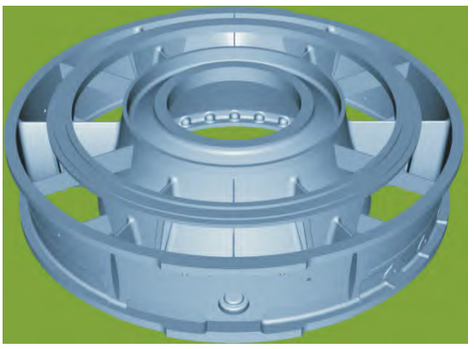


图 4 钛合金机匣模型
Fig.4 Titanium magazine model

2.2 结果分析

使用该钛合金铸件及对应的浇注系统，分别生成了 5 mm 网格、4 mm 网格、3 mm 网格和 1 mm 网格,如图 5~8 所示。数量级分别从百万级别到亿级别。网格化时间从开始导入 STL 模型开始计时,网格文件保存完毕时计时结束。钛合金机匣铸件模型 4次网格化测试的参数以及网格化结果如表 3 所示。

3 数值模拟

华铸 CAE 能够实现铸造充型过程、凝固过程、组织性能、应力应变及热处理过程的模拟[21]。使用华铸 CAE 12.5 对该程序生成的 4 mm 网格进行凝固模拟分析,初始温度为 1 800 ℃。凝固模拟的结果如图 9 所示。结果显示,华铸 CAE 在该程序生成的网格文件上正确地进行了凝固模拟。

表3 模型相关设定参数和生成网格信息
Tab.3 Parameters set for the model and generating grid information

Test number	Voxel size	Voxel number	Time	Result
1	5 mm	4 943 120	1.667 s	Fig. 5
2	4 mm	9 526 248	1.960 s	Fig. 6
3	3 mm	22 049 918	2.730 s	Fig. 7
4	1 mm	572 148 990	14.629 s	Fig. 8

4 结论

(1)该算法生成的网格准确还原了模型的特征,可以用于进行数值计算。

(2)该算法生成 4 943 120 个 5 mm 网格用时 1.667 s;生成 9 526 248 个 4 mm 网格用时 3.442 s;生成 22 049 918 个 3 mm 网格用时 4.353 s;生成 572 148 990 个 1 mm 网格用时 14.629 s;该程序具备快速实现网格化的能力。

(3)使用华铸 CAE12.5 对该程序生成的 4 mm 网格进行凝固模拟,模拟过程正确执行,该算法生成的网格可以用于数值模拟。

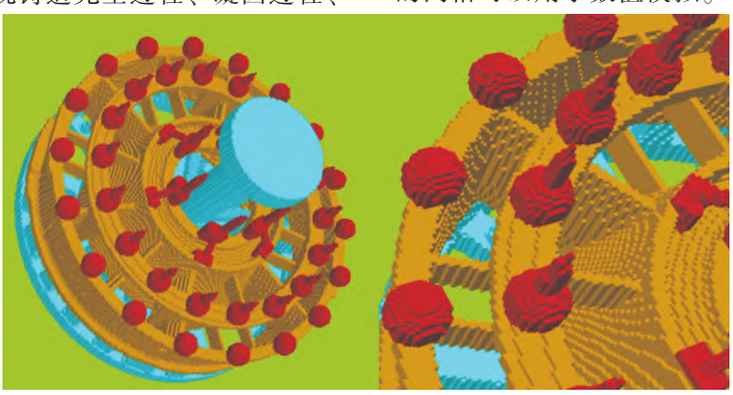


图 5 网格化结果(5 mm)
Fig.5 Mesh result (5 mm)

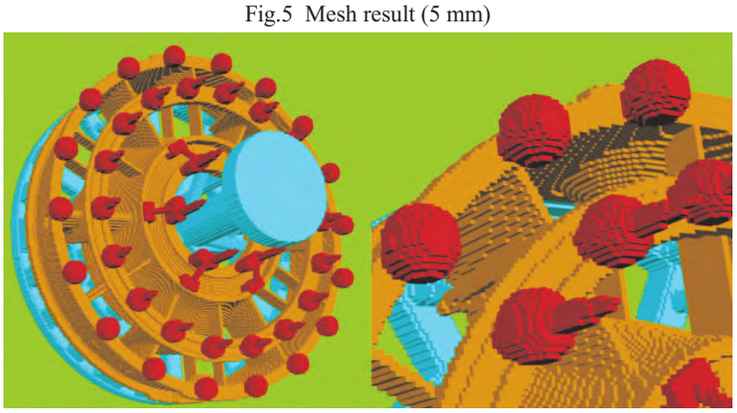


图 6 网格化结果(4 mm)
Fig.6 Mesh result (4 mm)

中国知网 https://www.cnki.net

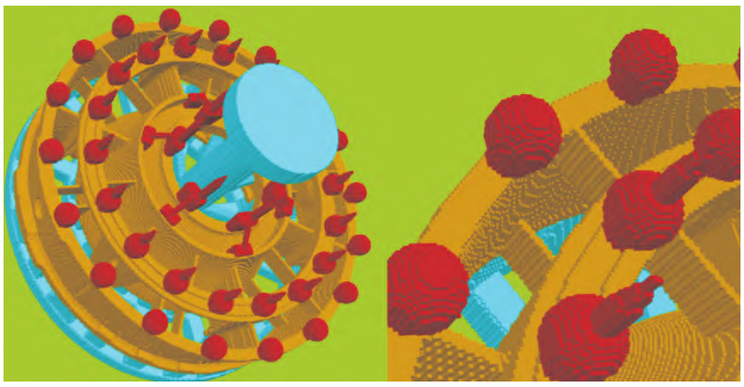


图 7 网格化结果(3 mm)
Fig.7 Mesh result (3 mm)

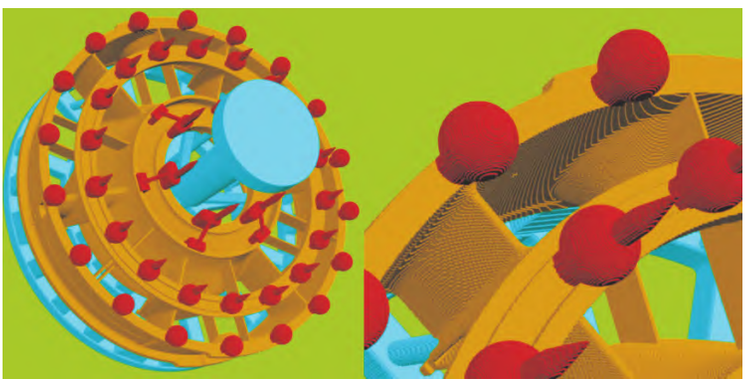


图 8 网格化结果(1 mm)
Fig.8 Mesh result (1 mm)

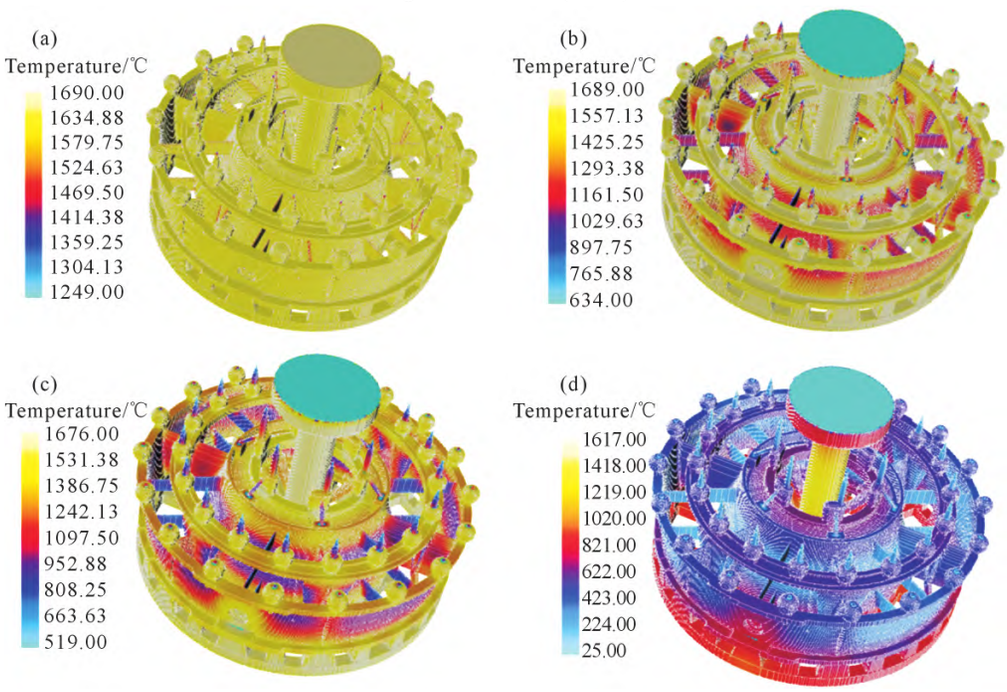


图 9 凝固模拟结果:(a) 凝固经历时间 3.65 s;(b) 凝固经历时间 39.74 s;(c) 凝固经历时间 171.98 s;(d) 凝固经历时间 1 205.92 s
Fig.9 Solidification simulation results: (a) after 3.65 s of solidification; (b) after 39.74 s of solidification; (c) after 171.98 s of solidification; (d) after 1 205.92 s of solidification

(4)如果多线程方式或者结合使用 GPU 实现该算法,执行时间将会进一步减少。

参考文献:

[1] 金和喜,魏克湘,李建明,周建宇,彭文静. 航空用钛合金研究进展[J]. 中国有色金属学报,2015, 25(2): 280-292.
JIN H X, WEI K X, LI J M, ZHOU J Y, PENG W J. Research development of titanium alloy in aerospace industry[J]. The Chinese Journal of Nonferrous Metals, 2015, 25(2): 280-292.

[2] 肖树龙,陈玉勇,朱洪艳,田竟,吴宝昌. 大型复杂薄壁钛合金铸件熔模精密铸造研究现状及发展[J]. 稀有金属材料与工程,

中国知网 https://www.cnki.net

2006, 35(5): 678-681.
XIAO S L, CHEN Y Y, ZHU H Y, TIAN J, WU B C. Recent advances on precision casting of large thin wall complex castings of titanium alloys[J]. Rare Metal Materials and Engineering, 2006, 35 (5): 678-681.

[3] 朱知寿. 我国航空用钛合金技术研究现状及发展 [J]. 航空材料学报,2014, 34(4): 44-50.
ZHU Z S. Recent research and development of titanium alloys for aviation application in China[J]. Journal of Aeronautical Materials, 2014, 34(4): 44-50.

[4] 张美娟,南海,鞠忠强,高富辉,郄喜望,朱郎平. 航空铸造钛合金及其成型技术发展[J]. 航空材料学报,2016, 36(3): 13-19.
ZHANG M J, NAN H, JU Z Q, GAO F Q, QIE X W, ZHU L P. Aeronautical cast Ti alloy and forming technology development[J]. Journal of Aeronautical Materials, 2016, 36(3): 13-19.

[5] 宋浩,韩冬,赵军,刘时兵,史昆,徐凯,刘鸿羽,李重阳. 钛合金熔模精密铸造技术的发展现状 [J]. 铸造,2020, 69 (12): 1304-1311.
SONG H, HAN D, ZHAO J, LIU S B, SHI K, XU K, LIU H Y, LI C Y. Development status of Ti alloy investment casting technology[J]. Foundry, 2020, 69(12): 1304-1311.

[6] 谭云骧,马聚怀,许庆彦. 铝合金低压铸造过程数值模拟[J]. 铸造,2024, 73(3): 270-281.
TAN Y X, MA J H, XU Q Y. Numerical simulation of low pressure die casting process of aluminum alloy[J]. Foundry, 2024, 73 (3): 270-281.

[7] 徐琴,吴士平,薛祥. 铸造过程数值模拟中的网格技术[J]. 铸造设备与工艺,2010(2): 40-45.
XU Q, WU S P, XUE X. Grid generation technique in numerical simulation on casting process[J]. Foundry Equipment and Technology, 2010(2): 40-45.

[8] 曹流. 铸造充型过程中成形类缺陷演化机理及数值模拟研究[D]. 武汉:华中科技大学,2019.
CAO L. Study on the evolution mechanism and numerical simulation of forming defects during cast-filling processes[D]. Wuhan: Huazhong University of Science and Technology, 2019.

[9] 周建兴,刘瑞祥,陈立亮,林汉同. 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.
ZHOU J X, LIU R X, CHEN L L, LIN H T. Research on ray piercing method based on STL in mesh division[J]. Foundry Technology, 2001(1): 15-17.

[10] 梁英业,戴挺,赵建新,朱鸣芳. 基于 STL 的切片线扫描法网格剖分技术[J]. 铸造,2005, 54(10): 64-67.
LIANG Y Y, DAI T, ZHAO J X, ZHU M F. Mesh generation using a slice line scanning method based on STL files[J]. Foundry, 2005, 54(10): 64-67.

[11] 吴晓军,刘伟军,王天然. 基于八叉树的三维网格模型体素化方法[J]. 工程图学学报,2005, 26(4): 1-7.
WU X J, LIU W J, WANG T R. Octree structure based voxelization of polygonal meshes[J]. Journal of Graphics, 2005, 26(4): 1-7.

[12] DONG Z, CHEN W, BAO H J, ZHANG H X, PENG Q S. Real-time voxelization for complex polygonal models[A]. 12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings[C]. Seoul: IEEE, 2004. 43-50.

[13] 齐龙,肖素梅,刘云楚,廖玲玲,蔡云龙. 基于 GPU 的并行非结构网格生成技术研究[J]. 机械设计与制造,2013(2): 184-186.
QI L, XIAO S M, LIU Y C, LIAO L L, CAI Y L. Research on parallel unstructured mesh generation technology based on GPU[J]. Machinery Design & Manufacture, 2013(2): 184-186.

[14] 戴愿桥. 基于八叉树的铸造前处理自适应网格划分算法研究及应用[D]. 武汉:华中科技大学,2020.
DAI Y Q. The Research and application of adaptive meshing algorithm of pre-casting processing based on octree[D]. Wuhan: Huazhong University of Science and Technology, 2020.

[15] 陈涛,廖敦明,庞盛永,周建新. 复杂铸造工艺装配体有限元网格划分关键技术研究[J]. 铸造设备与工艺,2014(5): 16-20.
CHEN T, LIAO D M, PANG S Y, ZHOU J X. Study on the finite element mesh generation method for complicated casting process [J]. Foundry Equipment and Technology, 2014(5): 16-20.

[16] 殷亚军. 基于八叉树网格技术的相场法金属凝固过程组织模拟的研究[D]. 武汉:华中科技大学,2014.
YIN Y J. Research on the microstructure simulation of the alloy solidification based on phase field method by the octree mesh technology[D]. Wuhan: Huazhong University of Science and Technology, 2014.

[17] 毕林,王李管,陈建宏,冯兴隆. 基于八叉树的复杂地质体块段模型建模技术[J]. 中国矿业大学学报,2008, 37(4): 532-537.
BI L, WANG L G, CHEN J H, FENG X L. Study of octree-based block model of complex geological bodies[J]. Journal of China University of Mining & Technology, 2008, 37(4): 532-537.

[18] 戴愿桥,廖敦明,陈涛,陈航,曹流. 基于八叉树的自适应网格剖分算法研究[A]. 2017 中国铸造活动周论文集[C]. 苏州:中国机械工程学会,2017. 5.
DAI Y Q, LIAO D M, CHEN T, CHEN H, CAO L. An algorithm research of adaptive mesh generation based on octree[A]. Proceedings of Foundry Week 2017 [C]. Suzhou: Chinese Mechanical Engineering Society, 2017. 5.

[19] 林菲,邹玲,张聪. 基于混合层次包围盒的快速碰撞检测算法[J]. 计算机仿真,2023, 40(9): 454-457.
LIN F, ZOU L, ZHANG C. Fast collision detection algorithm based on hybrid hierarchical bounding box[J]. Computer Simulation, 2023, 40(9): 454-457.

[20] AKENINE-MÖLLSER T. Fast 3D triangle-box overlap testing[J]. Journal of Graphics Tools, 2001, 6(1): 29-33.

[21] 殷亚军. 华铸 CAE 新进展及其应用[A]. 2019 中国铸造活动周论文集[C]. 武汉:中国机械工程学会,2019. 470.
YIN Y J. New progress and application of InteCAST[A]. Proceedings of China Foundry Congress 2019[C]. Wuhan: Chinese Mechanical Engineering Society, 2019. 470.

中国知网 https://www.cnki.net