

DOI:10.16356/j.1005-2615.2023.01.010

三维网格动态细化技术在切削仿真中的应用

杨 振, 王 洋, 王禹封, 苏宏华

(南京航空航天大学机电学院, 南京 210016)

摘要:为解决切削加工有限元仿真模型求解速度与精度难以平衡的问题,提出了一种适用于三维有限元模型的网格动态细化算法。该算法的主要功能包括网格细化区域判断、网格细化以及新旧网格物理场传递。采用 Python 语言对 Abaqus 软件进行二次开发,将该算法应用于 Ti2AlNb 钛合金车削加工仿真之中,并最终通过实验验证了仿真模型的准确性。与采用局部网格细化的仿真模型计算结果对比,采用网格动态细化技术的仿真模型求解切削力误差增加了 7.3%,求解最大应力误差增加 0.2%,求解速度提升 174.2%。实现了在保证仿真计算精度的基础上,有效提高运算速度。

关键词:三维网格动态细化;物理场传递;金属切削;切削仿真

中图分类号:TG501 **文献标志码:**A **文章编号:**1005-2615(2023)01-0080-09

Application of 3D Mesh Dynamic Refinement Technology in Cutting Simulation

YANG Zhen, WANG Yang, WANG Yufeng, SU Honghua

(College of Mechanical & Electrical Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China)

Abstract: Since it is difficult to balance the speed and accuracy of the finite element simulation model of cutting processing, a dynamic mesh refinement algorithm for 3D finite element model is proposed. The functions of the algorithm include mesh refinement area judgment, mesh refinement and transfer of physics fields between new and old meshes. The algorithm is applied to the turning simulation of Ti2AlNb with the secondary development of Abaqus using Python, and the accuracy of the simulation model is verified through experiments. Compared with the simulation results of the local mesh refinement, the simulation using dynamic mesh refinement increases 7.3% in cutting force solution error and increases 0.2% in maximum stress solution error, with an increased 174.2% solution speed. The simulation operation speed is effectively improved while ensuring the calculation accuracy.

Key words: 3D mesh dynamic refinement; physical field transfer; metal cutting; cutting simulation

随着计算机技术与数值计算方法的发展,有限元法已成为解决工程问题的重要方法之一。作为一种近似分析方法,有限元计算求解的精度受网格数量、网格形状、网格单元类型和求解方法等多重因素的影响,其中模型网格的数量与质量对于求解准确度有着最为直接的影响^[1-3]。但网格数量增加的同时也伴随着计算量的增加,因此通常会对仿真

模型采用网格动态细化的方法,将仿真对象中几何结构较为复杂,物理场突变较大的区域的网格进行细化,从而在获得理想计算精度的情况下减少求解计算消耗的时间^[4]。

由于采用自适应网格划分的模型相对于传统网格划分的模型在求解的精度、速度和收敛性方面均有了明显的提升,所以在具有多物理场耦合作用

收稿日期:2022-05-25;**修订日期:**2022-10-14

通信作者:苏宏华,男,教授,博士生导师,E-mail:shh@nuaa.edu.cn。

引用格式:杨振,王洋,王禹封,等. 三维网格动态细化技术在切削仿真中的应用[J]. 南京航空航天大学学报,2023,55(1):80-88. YANG Zhen, WANG Yang, WANG Yufeng, et al. Application of 3D mesh dynamic refinement technology in cutting simulation[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2023, 55(1):80-88.

的仿真模型之中得到了广泛的应用^[5-6],例如制造领域中的切削加工模型。Baker等^[7]运用二维网格自适应技术,在对钛合金 Ti6Al4V 在切削加工过程中发生的绝热剪切现象进行了仿真研究,并将刀尖位置的网格进行了加密,最终有效缩短了仿真计算的时间,并得出了随着切削速度的增加,绝热剪切带不断减小的结论。Labergère等^[8-9]将网格自适应技术运用至二维切削仿真模型中,提高了模型整体的仿真效率与刀具切削区域的仿真结果精度。李宗旺^[10]对二维四边形网格动态细化方法进行了研究,将其应用在 Ti6Al4V 的二维切削仿真之中,最终得出采用网格动态细化方法的模型相较于常规网格划分模型,在求解结果的应力大小与分布情况基本相同的情况下,运算速度提高了 210%。

在现有的研究中,自适应网格细化的对象主要为三角形网格或四面体网格,而在实际的仿真过程中,四边形与六面体网格在计算时有着更好的精度以及更快的计算速度。因此本文针对六面体网格有限元模型,从网格数量对模型计算精度与速度影响的角度出发,面向 Abaqus 软件,以 Python 为编程语言,针对切削加工仿真过程,研究了三维网格动态细化相关算法,主要包括:(1)用于进行网格细化区域以及网格单元删除判定的网格细化区域判断方法;(2)用于在原始网格单元内部插入新的节点,并连接形成新的单元的网格细化算法;(3)对网格间悬挂节点进行判断并约束的疏密网格过渡算法;(4)实现网格细化前后模型物理场传递,从而确保仿真连续性的物理场传递方法。最终建立了采用 3 种网格划分方式的 Ti2AlNb 钛合金切削加工仿真模型,并依据实验测得的切削力数值,将仿真结果与实验值对比分析,验证了网格动态细化技术在提高仿真运算效率方面的有效性。

1 三维网格动态细化技术研究

1.1 网格细化区域判断

在对有限元模型进行网格动态细化前,首先需要指定网格细化的判断准则,以确定出哪些网格需要进行细化^[11]。在切削加工的过程中,工件上的应力与应变主要集中于刀具附近材料去除的位置,因此网格细化区域应设置为刀具切削材料的位置及其周边一定范围内的网格^[12]。根据此特点,本文网格细化区域的判断方法选用几何位置判断法。

在切削加工有限元仿真中,为便于网格动态细化区域的判断,需要对刀具运动轨迹进行离散处

理,将加工轨迹离散为独立分布的点。因此本文设计了针对刀具前进直线轨迹的离散方法,离散效果如图 1 所示。直线刀轨从起点至终点被依次分割为一系列间距为 d 的刀位点,当点的距离与刀轨终点的距离不足 d 时,则直接取终点为最后一个刀位点。



图 1 直线刀轨离散示意图

Fig.1 Schematic diagram of the linear tool path discrete method

完成对刀轨的离散处理之后,需要根据得到的刀位点,对所需细化单元进行判断选择,因此本文制定了工件网格模型节点与给定几何参数长方体空间关系的判断准则。空间中的点与长方体区域示意如图 2 所示。图中长方体中心线与其底面交点为起点坐标,坐标为 (x_0, y_0, z_0) ,与顶面交点为终点坐标,坐标为 (x_1, y_1, z_1) ,平行于中心线的两对侧面与中心线的距离分别为 d_y, d_z 。长方体中心线方向向量为 $c(x_c, y_c, z_c)$, i 在 y_z 平面的投影与 y 轴正方向夹角为 α , c 与 x 轴正方向的夹角为 β 。

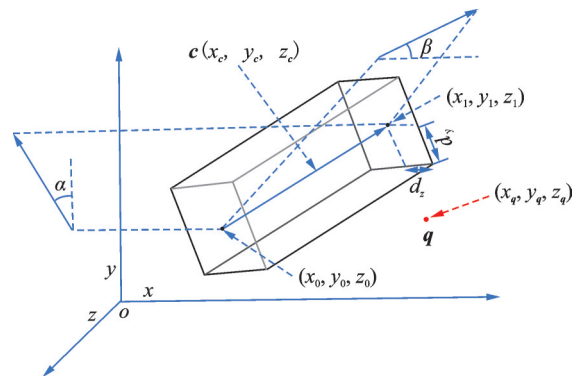


图 2 长方体区域示意图

Fig.2 Schematic diagram of cuboid area

该方法首先将长方体与需要判断的点进行平移与旋转变换,使得长方体的中心线方向与坐标轴 x 方向重合,此时长方体在 xy 平面内的投影如图 3 所示。最终将变换后的点坐标与长方体的边界值进行比较,则可实现对空间中任意点与长方体位置关系的判断。

1.2 三维网格细化方法

有限元方法中,网格单元是通过节点按照一定的顺序连接而成的,因此要进行网格的细化,需要首先通过模型原始粗网格的节点生成用于构建新细网格单元的新节点。本文选用的细化类型为将单个网格分为 8 个网格,如图 4 所示。

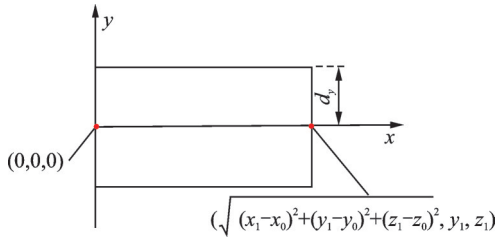


图3 长方体变换后xy平面投影

Fig.3 Projection on the xy plane of cuboid after transformation

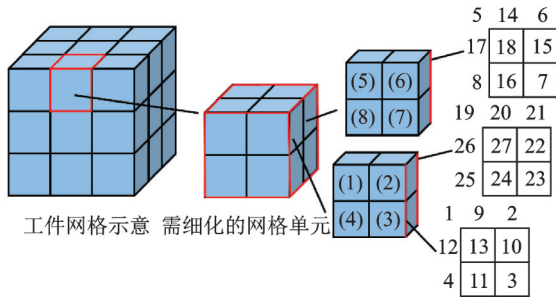


图4 原始模型粗网格细化方式

Fig.4 Coarse mesh refinement of original model

该细化方法的具体步骤为:

(1)读取工件网格模型的单元与节点信息。其中单元信息包含单元编号以及组成该单元的节点编号;节点信息包含节点编号与节点的坐标值。

(2)依据由几何位置判断法判断出的需要细化的单元与坐标信息,逐个在单元内插入新的节点。新生成节点的坐标计算方法为

$$\begin{bmatrix} x_9 & y_9 & z_9 \\ x_{10} & y_{10} & z_{10} \\ x_{11} & y_{11} & z_{11} \\ x_{12} & y_{12} & z_{12} \\ x_{13} & y_{13} & z_{13} \\ x_{14} & y_{14} & z_{14} \\ x_{15} & y_{15} & z_{15} \\ x_{16} & y_{16} & z_{16} \\ x_{17} & y_{17} & z_{17} \\ x_{18} & y_{18} & z_{18} \\ x_{19} & y_{19} & z_{19} \\ x_{20} & y_{20} & z_{20} \\ x_{21} & y_{21} & z_{21} \\ x_{22} & y_{22} & z_{22} \\ x_{23} & y_{23} & z_{23} \\ x_{24} & y_{24} & z_{24} \\ x_{25} & y_{25} & z_{25} \\ x_{26} & y_{26} & z_{26} \\ x_{27} & y_{27} & z_{27} \end{bmatrix} = \frac{1}{8} \begin{bmatrix} 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 4 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 \\ 2 & 0 & 0 & 2 & 2 & 0 & 0 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \\ x_5 & y_5 & z_5 \\ x_6 & y_6 & z_6 \\ x_7 & y_7 & z_7 \\ x_8 & y_8 & z_8 \end{bmatrix} \quad (1)$$

设图4中工件网格需要细化的单元编号为n, 右侧1号位置节点的坐标为(x₁,y₁,z₁),2号位置节点的坐标(x₂,y₂,z₂),以此类推至27号位置节点。其中1~8号位置的节点为原始网格节点,其坐标值由工件信息读取后已知。9~27号位置的节点为需要新生成的节点,其坐标需要通过原始网格节点坐标计算得出。

(3)对新生成的节点进行编号。设细化前工件网格节点编号的最大值为n₀,n_i为图4中第i号(i为网络节点上的编号)位置节点的编号,则n_i为

$$n_i = n_0 + i - 8 \quad i = 9, 10, \dots, 27 \quad (2)$$

(5)对细化后的新网格单元编号。设细化前工件网格单元编号的最大值为e₀,e_i为图4中第i号位置单元的编号,则

$$e_i = e_0 + i \quad i = 1, 2, \dots, 8 \quad (3)$$

(6)确定新生成单元与节点的包含关系。由图4可知,新生成的单元编号e_i与其内部所包含的节点编号n_i为

$$\begin{cases} e_1 = \{n_1, n_9, n_{12}, n_{13}, n_{19}, n_{20}, n_{26}, n_{27}\} \\ e_2 = \{n_2, n_9, n_{10}, n_{13}, n_{20}, n_{21}, n_{22}, n_{27}\} \\ e_3 = \{n_3, n_{10}, n_{11}, n_{13}, n_{22}, n_{23}, n_{24}, n_{27}\} \\ e_4 = \{n_4, n_{11}, n_{12}, n_{13}, n_{24}, n_{25}, n_{26}, n_{27}\} \\ e_5 = \{n_5, n_{14}, n_{17}, n_{18}, n_{19}, n_{20}, n_{26}, n_{27}\} \\ e_6 = \{n_6, n_{14}, n_{15}, n_{18}, n_{20}, n_{21}, n_{22}, n_{27}\} \\ e_7 = \{n_7, n_{15}, n_{16}, n_{18}, n_{22}, n_{23}, n_{24}, n_{27}\} \\ e_8 = \{n_8, n_{16}, n_{17}, n_{18}, n_{24}, n_{25}, n_{26}, n_{27}\} \end{cases} \quad (4)$$

以上步骤为针对工件原始粗网格中的单个网格进行细化,若需对某一区域的多个网格进行细化,通过对单个单元循环使用此方法即可实现。

1.3 疏密网格过渡方法

有限元模型中,若相邻网格公共边上的节点未与其他节点相连,则称该节点为悬挂节点^[13],如图5中的p点。三维网格细化之后产生的悬挂节点按照其与原粗网格的相对位置,可以被分为两个类型,位于六面体网格单元边上的悬挂节点(图5(a))与位于单元面上的悬挂节点(图5(b))。在下文中分别称其为边上悬挂节点与面上悬挂节点。悬挂节点会极大增加模型求解的误差。对于悬挂节点的处理,本文采用约束等式法,主要利用悬挂节点周围的粗网格节点对悬挂节点施加约束,从而限制悬挂节点的自由度,实现节点位移值的准确求解。

对于图5(a)中的边上悬挂节点p,可以利用其两侧点a与b进行约束,将点p在进行有限元求解的过程中位移值为点a与点b插值得出,插值方法

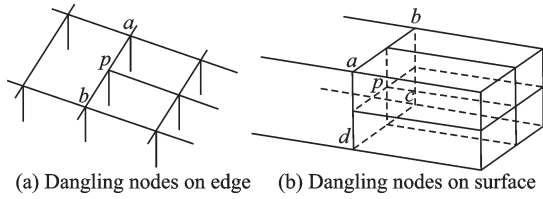


图 5 悬挂节点
Fig.5 Dangling nodes

选为线性插值法。对于图 5(b)中面上的悬挂节点 p 则需要利用与其同平面的 4 个点 a, b, c, d 进行约束,插值方法选用双线性插值法。借助线性与双线性插值的方法,悬挂节点的自由度能够受到指定节点的约束,从而实现求解中物理量在节点之间的正确传递。

在判断边上悬挂节点时,采用的方法是计算需要判断节点与悬挂节点理论位置的距离,当距离小于指定值时,则此点位于悬挂节点区域内,可被认为是悬挂节点。如图 6(a)所示,点 a, b 分别为细化前粗网格单元边上的两个节点,坐标分别为 $(x_a, y_a, z_a), (x_b, y_b, z_b)$,点 p 为此边悬挂节点的理论位置。由于在网格细化的过程中采用的网格划分方法为 1 个单元等分为 8 个单元,因此点 p 坐标为

$$(x_p, y_p, z_p) = \left(\frac{x_a + x_b}{2}, \frac{y_a + y_b}{2}, \frac{z_a + z_b}{2} \right) \quad (5)$$

点 m 为需要判断的点,坐标为 (x_m, y_m, z_m) 。当点 m 满足

$$\sqrt{(x_m - x_p)^2 + (y_m - y_p)^2 + (z_m - z_p)^2} < 0.1r \quad (6)$$

则认为点 m 为边上悬挂节点。其中 r 为细化前原始网格边长的最小值。采用这种判断方法是由于在进行求解后,模型的网格会发生一定的变形,导致悬挂节点的真实位置与理论位置会产生一些偏差,因此在进行判断时需要设定一定的允许误差。经验证 0.1 倍的最小原始网格边长可以确保判断的准确性。

在判断面上悬挂节点时,判断方法与边上悬挂节点基本相同。如图 6(b)所示,点 a, b, c, d 分别为细化前原始网格单元面上的 4 个节点,坐标分别为 $(x_a, y_a, z_a), (x_b, y_b, z_b)$,以此类推,理论悬挂节点 p 的坐标为

$$(x_p, y_p, z_p) = \left(\frac{x_a + x_b + x_c + x_d}{4}, \frac{y_a + y_b + y_c + y_d}{4}, \frac{z_a + z_b + z_c + z_d}{4} \right) \quad (7)$$

当点 m 坐标满足式(6)时,则认为点 m 为面上悬挂节点。

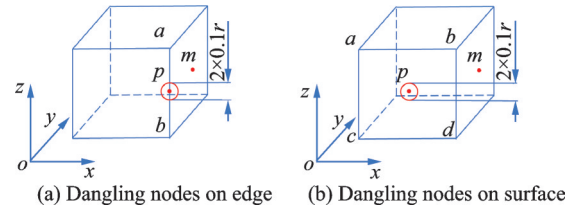


图 6 悬挂节点判断
Fig.6 Judgment of dangling nodes

上述方法可以实现对单个节点是否为悬挂节点进行判断,使用该方法对模型中所有新生成的节点进行判断,即可实现对模型所有悬挂节点的筛选。

1.4 物理场传递方法

在对网格模型进行细化处理后,新生成的网格模型与初始网格模型相比,网格单元与节点的数量发生了变化。为保证最终求解过程的准确性,需要将细化前网格模型求解所得的物理场信息以初始条件的形式,传递施加至新生成的网格模型之中。在仿真中较为重要的物理量有应力、塑性应变和温度等。本文中这些物理量传递所采用的方法基本相同,以下以应力为代表阐述物理场的传递方法。

物理场传递依靠的是粗网格与细网格之间建立的对对应关系。原始网格模型进行细化后,粗网格被切分成 8 个新的细网格,并最终得到了式(4)的对应关系。因此本文选用了八叉树法对网格单元的物理场信息进行存储与传递。在对应力场信息进行存储时,可以依据上述八叉树结构,以单个单元的应力值为节点,建立多级网格细化模型的应力场信息八叉树群,如图 7 所示。第 2 级网格单元的应力值为第 1 级网格单元的子节点,第 3 级网格单元的应力值为第 2 级的子节点,以此类推。

当网格细化完成后,需要确定新生成的网格单元应当传递的应力值大小时,只需要依据该单元应

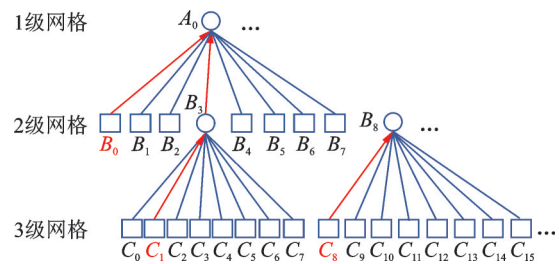


图 7 网格细化模型应力场存储结构

Fig.7 Stress field storage structure of mesh refinement model

力值在应力场八叉树中存储的节点位置,向上找出该子节点所在八叉树的根节点,该根节点的应力值即为此单元初始条件应当施加的应力值。例如当需要确定图7中 B_0 子节点位置所对应的应力值时,则沿 B_0 所在八叉树向上寻找根节点,可得根节点为 A_0 ,所以 B_0 子节点的应力值应当为 A_0 根节点的应力值。同理可得, C_1 子节点的应力值为 A_0 根节点的应力值, C_8 子节点的应力值为 B_8 根节点的应力值。

获得了每1子节点所对应单元的应力值之后,使用Abaqus中的预定义场功能,在inp文件中以*initial conditions关键字的形式即可实现对模型预定义场的施加。

1.5 网格动态细化仿真模型建立流程

为在实际仿真模型中应用网格动态细化算法,本文通过Python语言对上述算法的功能与算法之间的衔接进行了实现,仿真模型建立的具体流程如图8所示,建模的具体步骤为:

(1) 提供前处理插件的输入,包括由工件Abaqus模型生成的inp文件,由UG生成的cls加工刀轨文件,以及用户在前处理插件界面中输入的仿真相关参数。

(2) 对刀轨进行离散处理,依据离散得到的刀位点判断需要进行细化的网格。

(3) 对细化后的模型进行悬挂节点约束,并进行施加切削力,选择切削单元等操作。

(4) 依据刀位点在整个加工刀轨中的位置,判断是否进行后续的物理场传递以及仿真结束操作。

在这些步骤中,用户需要手动进行的是第1步中的工作,即在程序运行之前,准备用于提供输入的inp文件与cls文件。inp文件的生成需要将工件的几何模型导入至Abaqus中,完成有限元建模的基本流程,如材料设置,网格划分等,后续通过作业模块的写入输入文件即可生成。cls文件的生成借助UG软件的加工模块实现的,将几何模型导入至UG中,选择UG中的加工模块,确定加工刀具信息,加工方式与加工参数,最终对加工过程进行仿真,生成刀轨文件。

完成inp文件于cls文件的生成工作后,将文件存储在与程序相同的目录下,程序即可自动进行仿真模型的建立。

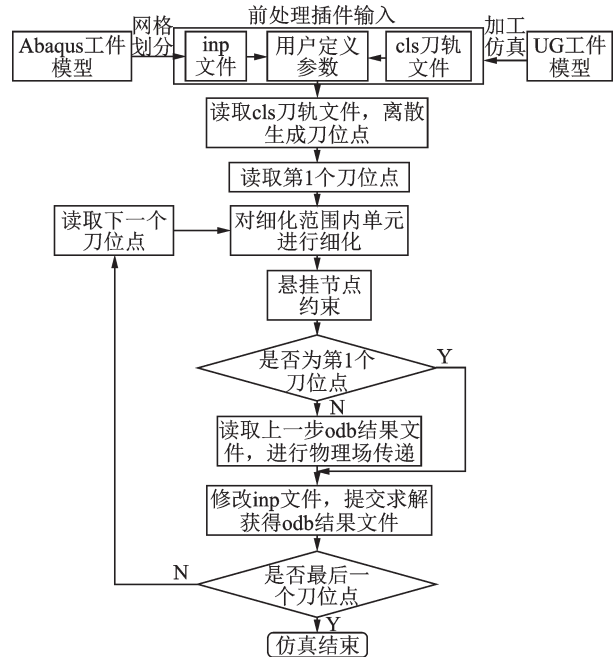


图8 切削仿真建模流程图

Fig.8 Cutting simulation modeling flowchart

2 切削仿真模型的建立与结果

2.1 切削仿真有限元模型建立

为验证三维网格动态细化技术在仿真中的效果,本文建立了考虑切屑分离的正交车削三维有限元仿真模型,如图9所示。

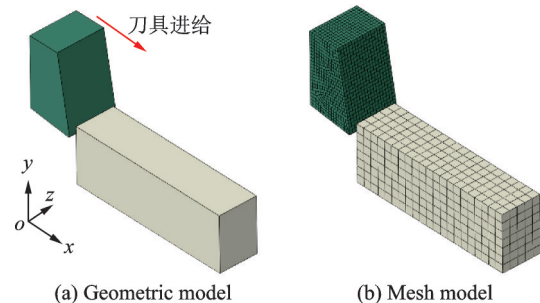


图9 车削仿真模型

Fig.9 Turning simulation model

其中刀具材料为硬质合金,工件材料为Ti2Al-Nb钛合金,本构选用采用修正后的TANH模型^[14],参数如表1所示。刀具设置为刚体,刀尖圆弧半径为0.03 mm,刀具前角为 5° ,后角 7° ,刀具单元类型为C3D8R,网格数为6 069个。工件尺寸为 $1.68\text{ mm} \times 0.56\text{ mm} \times 0.3\text{ mm}$,网格数为735个,单元类型为结构化一阶六面体单元C3D8R。边界条件:刀具移动速度方向为沿x正方向,工件底端固定,限制其6个自由度。初始条件:温度 20°C ,忽略与环境的对流换热影响。摩擦模型选用Zorev模

型^[15]。切削参数为:切削速度 v , 50 m/min;进给量 f , 0.1 mm/r;切削宽度 a_w , 0.3 mm。

表 1 工件与刀具主要材料参数^[14]

材料性能	工件	刀具
密度/($\text{kg}\cdot\text{m}^{-3}$)	5 350	15 000
热导率/($\text{W}\cdot(\text{m}\cdot\text{K})^{-1}$)	6.3	46
弹性模量/GPa	120.8	800
泊松比	0.31	—
比热容/($\text{J}\cdot(\text{kg}\cdot\text{K})^{-1}$)	431	203
热膨胀系数/ K^{-1}	8.22×10^{-6}	4.7×10^{-6}
室温断裂韧性/($\text{MPa}\cdot\text{m}^{1/2}$)	39	—

2.2 网格动态细化仿真结果

在上述仿真条件下,采用网格动态细化的三维车削仿真结果如图 10 所示。

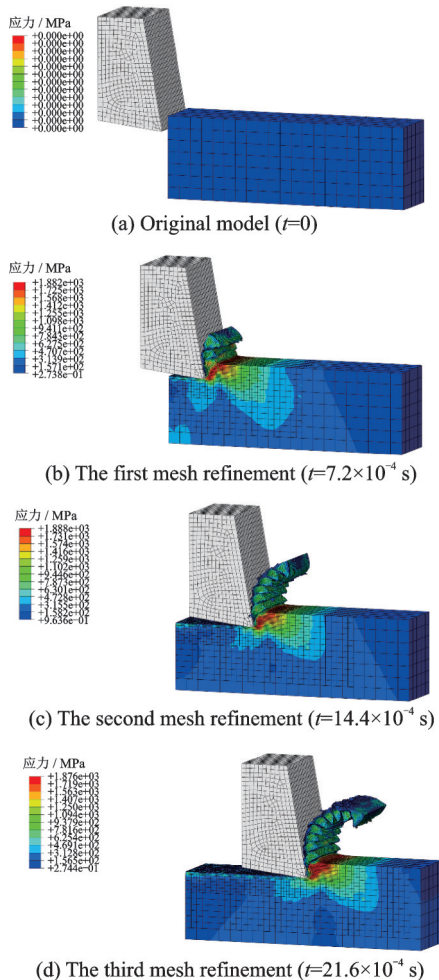


图 10 网格动态细化切削仿真结果

Fig.10 Simulation results of mesh dynamic refinement cutting

图 10(a~d)分别为切削时长 0, 7.2×10^{-4} , 14.4×10^{-4} 和 21.6×10^{-4} s时的应力分布云图。可以看出在仿真过程中,随着刀具的移动,刀尖与其

附近区域的网格得到了细化,其余位置的网格则保持了原始的尺寸。由于对模型的悬挂节点进行了约束,疏密网格的应力场实现了平滑的过渡。与此同时,仿真中得到的切屑形态为锯齿状,这与通常实验中切削钛合金得到的锯齿状切屑相吻合。

3 切削实验验证与仿真结果对比

3.1 切削实验条件

本课题组何临江^[14]前期进行了 Ti2AlNb 钛合金的车削实验。实验在 SK-50P 数控车床上进行。刀具选用涂层硬质合金刀具,前角 5° ,后角 7° ,基体材料为 WC,粘结相为 Co,涂层材料为 TiAl 复合涂层。使用压电式测力仪与 DynoWare 信号采集系统对切削力进行测量。车削加工实验装置如图 11 所示,切削实验参数如表 2 所示。

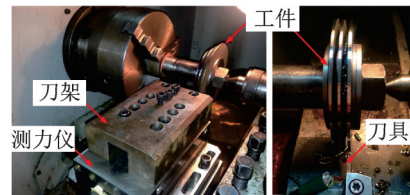


图 11 车削加工实验装置^[14]

Fig.11 Turning experimental device^[14]

表 2 切削实验工艺参数

切削参数	数值
切削宽度 a_w / mm	0.3
切削速度 v / ($\text{m}\cdot\text{min}^{-1}$)	50
进给量 f / ($\text{mm}\cdot\text{r}^{-1}$)	0.1

3.2 切削力实验值与仿真结果对比

在 3.1 节所述的实验条件下,最终测得加工过程中的切削力如表 3 所示。

表 3 实验测得切削力数值

切削力	数值
F_z	102
F_x	66
F_y	42

在加工过程中主切削力 F_z 数值较大,变化较为明显,因此本文以其为对象进行仿真与实验结果的对比。网格动态细化模型仿真过程中的主切削力变化情况如图 12 所示。

由于在车削过程中形成的切屑为锯齿状,所以可以看出主切削力的数值出现了较为明显的波动,

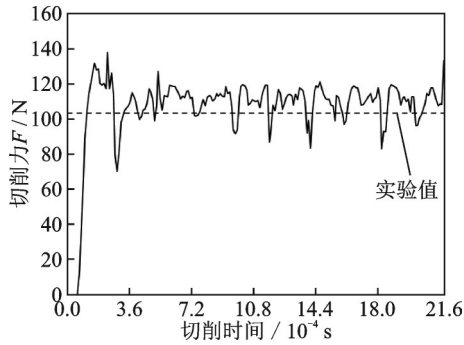


图12 切削仿真中主切削力大小

Fig.12 Value of main cutting force in cutting simulation

计算得其平均值为 114 N, 与实验值进行对比, 可知仿真误差为 11.7%, 从而说明该仿真模型是基本准确的。

3.3 不同网格划分方式模型仿真结果对比

为验证网格动态细化仿真模型在求解时的准确性与高效性。本文建立了3种采用不同网格划分方式的有限元模型对切削过程进行仿真, 分别为全局粗网格模型, 局部细网格模型与动态细网格模型, 如图13所示。其中动态细网格模型的网格会随着仿真中刀具的移动对刀尖及其附近的网格进行加密。

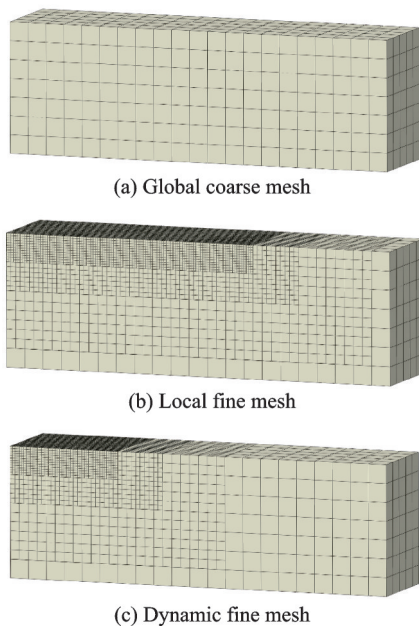


图13 不同网格划分方式仿真模型

Fig.13 Simulation models with different meshing methods

首先对3种仿真模型的切削力计算结果进行对比。分别提取其切削阶段的主切削力 F_z , 并计算其平均值以及与实验测量结果的误差, 结果如表4所示。从表4中可以得出, 与局部细网格模型的求解结果误差相比, 动态细网格模型的切削力仿真

结果误差小幅增加了7.3%; 全局粗网格模型的求解误差大幅增加了79.8%, 计算精度出现明显下降。这是由于在有限元仿真中, 网格数量对求解精度有直接的影响, 全局粗网格模型的网格数量最少, 因此求解结果误差较大。局部细网格与动态细网格模型的网格数量相对较多, 其中动态细网格模型的网格数量少于局部细网格模型, 但由于动态细网格模型在材料切削的区域保证有足够数量的网格, 因而两模型的仿真误差基本处于同一水平。

表4 不同网格划分方式模型切削力与误差

Table 4 Cutting force and error of models with different meshing methods

网格划分方式	切削力/N	误差/%
全局粗网格	122	19.6
局部细网格	113	10.9
动态细网格	114	11.7

其次对3种模型的应力求解结果进行对比。当切削仿真时长为 2×10^{-4} s时, 不同仿真模型的工件应力云图如图14所示。

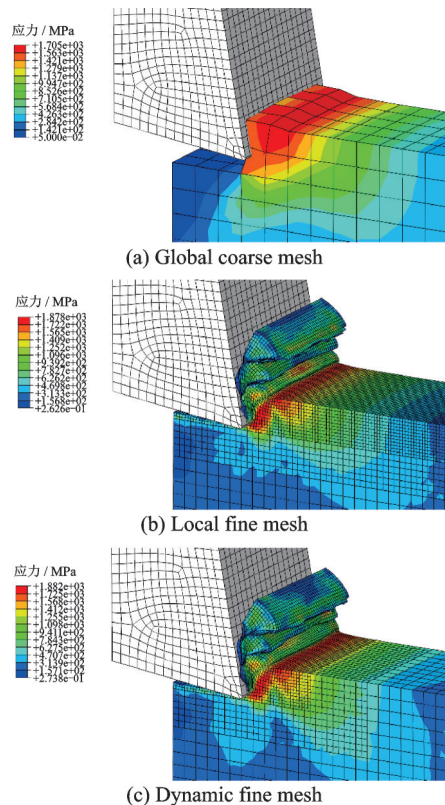


图14 不同网格划分方式仿真应力云图

Fig.14 Simulation models with different meshing methods

由图14可知, 全局粗网格模型由于网格数量较少, 在切削过程中没有产生与实际加工中类似的锯齿状切屑, 而局部细网格与动态细网格模型均还

原出了锯齿状的切屑形态。将局部细网格模型工件的应力分布与其余两种模型进行对比,可以看出全局粗网格模型的应力分布误差较大,而动态细网格模型的应力分布则与局部细网格模型基本一致。统计该时刻下 3 种模型的应力最大值,并以局部细网格模型的仿真结果为基准计算误差值,结果如表 5 所示。可知相较于局部细网格模型,动态细网格模型求解得到的最大应力误差仅为 0.2%,而全局粗网格模型的误差较大,为 9.2%,绝对值增加了 9%。

表 5 不同网格划分方式下模型最大应力值与误差

Table 5 Maximum stress value and error of models with different meshing methods

网格划分方式	最大应力/MPa	误差/%
全局粗网格	1 705	9.2
局部细网格	1 878	—
动态细网格	1 882	0.2

3.4 不同网格划分方式模型仿真速度对比

将 3 种模型的网格数量与计算消耗时间进行统计,计算平台配置为四核、3.2 GHz 主频处理器,16 GB 运行内存,结果如表 6 所示。

表 6 不同网格划分方式模型网格数量与求解时间

Table 6 Number of meshes and solution time of models with different meshing methods

网格划分方式	网格数量/个	求解时间/s
全局粗网格	735	874
局部细网格	69 804	42 121
动态细网格	30 555~69 804	15 364

由表 6 中数据可知全局粗网格模型由于网格数量最少,求解消耗时间也最短,为 874 s。局部细网格模型的求解时间最长,花费 42 121 s。采用了网格动态细化的模型,由于其网格数量是随着加工的进行动态增多,从 30 555 个逐渐增加至 69 804 个,相较于局部细网格模型的网格数量有了明显减少,因此求解花费时间减少至 15 364 s,为局部细网格的 36.5%,即求解速度相较于局部细网格模型提升了 174.2%。

综合考虑上述仿真结果与实验值的误差以及不同模型仿真所消耗时间,可以说明,网格动态细化技术可以在基本保证仿真精度的前提下,有效提高仿真计算速度。本文针对 Ti2AlNb 钛合金的车削加工仿真中,相较于局部细网格模型,采用网格动态细化技术的模型求解切削力误差略小幅增加

7.3%,求解最大应力误差略微增加 0.2%,求解速度提升了 174.2%。

4 结 论

本文针对切削加工有限元仿真模型求解速度与精度难以平衡的问题,从有限元网格数量对模型计算精度与速度影响的角度出发,面向 Abaqus 软件,以 Python 为编程语言,对三维网格动态细化的相关算法进行了研究,最终将其应用于 Ti2AlNb 钛合金的车削加工仿真之中,具体研究成果如下:

(1)研究了三维网格动态细化相关算法,主要包括网格细化区域判断方法;对原始粗网格进行细化的三维网格细化方法;处理细化后悬挂节点处物理量求解不准确的疏密网格过渡算法;确保仿真连续性,实现新旧网格模型求解结果传递的物理场传递方法。

(2)建立了采用 3 种网格划分方式的 Ti2AlNb 钛合金车削加工仿真模型,对比车削实验测得的切削力数据,验证了仿真模型的准确性。并将采用不同网格划分方式的仿真计算结果对比后得出结论:采用网格动态细化技术的仿真模型相较于局部细网格模型,求解切削力误差小幅增加 7.3%,求解最大应力误差略微增加 0.2%,求解速度大幅提升 174.2%。验证了网格动态细化技术在保证仿真精度的同时,能够有效提高仿真求解效率。

参考文献:

- [1] WILSON E L. Automation of the finite element method: A personal historical view[J]. Finite Element Analysis Design, 1993, 13: 91-104.
- [2] THACKER W C. A brief review of techniques for generating irregular computational grids[J]. International Journal for Numerical Methods in Engineering, 1980, 15(9): 1335-1341.
- [3] 王勖成. 有限单元法[M]. 北京: 清华大学出版社, 2003: 162-186.
WANG Xucheng. Finite element method[M]. Beijing: Tsinghua University Press, 2003: 162-186.
- [4] YANG F T, RASSINEUX A, LABERGERE C. A 3D h-adaptive local remeshing technique for simulating the initiation and propagation of cracks in ductile materials[J]. Computer Methods in Applied Mechanics and Engineering, 2018, 330: 102-122.
- [5] 陈涛. 自适应网格划分的有限单元法在土坡稳定中

- 的应用[D]. 杭州: 浙江大学, 2013.
- CHEN Tao. The application of adaptive remeshing with finite element method in the analyzation of slope stability[D]. Hangzhou: Zhejiang University, 2013.
- [6] JANSARI C, NATARAJAN S. Adaptive smoothed stable extended finite element method for weak discontinuities for finite elasticity[J]. *European Journal of Mechanics*, 2019, 78: 103824-103838.
- [7] BAKER M, ROSLER J, SIEMERS C. A finite element model of high speed metal cutting with adiabatic shearing[J]. *Computer and Structure*, 2002, 80(3): 495-513.
- [8] LABERGÈRE C, RASSINEUX A, SAANOUNI K. 2D adaptive mesh methodology for the simulation of metal forming processes with damage [J]. *International Journal of Material Forming*, 2011, 4: 317-328.
- [9] LABERGÈRE C, RASSINEUX A, SAANOUNI K. Numerical simulation of continuous damage and fracture in metal-forming processes with 2D mesh adaptive methodology[J]. *Finite Elements in Analysis & Design*, 2014, 82: 46-61.
- [10] 李宗旺. 基于 Abaqus 的二维切削仿真局部网格动态细化及前处理关键技术研究[D]. 南京: 南京航空航天大学, 2020.
- LI Zongwang. Research on dynamic refinement technology of 3D mesh in cutting simulation oriented to ABAQUS[D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2020.
- [11] 黄丽丽, 赵国群, 王忠雷. 栅格法三维六面体网格局部加密算法[J]. *计算机辅助设计与图形学学报*, 2010, 22(4): 612-618.
- HUANG Lili, ZHAO Guoqun, WANG Zhonglei. Grid-based local refinement algorithm of three-dimensional hexahedral meshes[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2010, 22(4): 612-618.
- [12] CAO Z Y, HE N, LI L. A finite element analysis of micromesoccale machining considering the cutting edge radius[J]. *Applied Mechanics ang Materials*, 2008, 11(12): 631-636.
- [13] JANSARI C, NATARAJAN S. Adaptive smoothed stable extended finite element method for weak discontinuities for finite elasticity[J]. *European Journal of Mechanics*, 2019, 78(103): 824-826.
- [14] 何临江. Ti2AlNb 金属间化合物切削加工基础研究, [D]. 南京: 南京航空航天大学, 2018.
- HE Linjiang. Fundamental research on machining of Ti2AlNb intermetallic alloys[D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2018.
- [15] WANG Y, SU H H, DAI J B, et al. A novel finite element method for the wear analysis of cemented carbide tool during high speed cutting Ti6Al4V process [J]. *The International Journal of Advanced Manufacturing Technology*, 2019, 103(2): 2795-2807.

(编辑:陈珺)