

工具経路生成時間短縮法の開発*

—加工点依存プログレッシブメッシュ表現の適用—

鬼頭亮太** 高杉敬吾*** 浅川直紀**** 岡田将人****

Development of Collision Check Method with Lower Calculation Cost for Tool Path Generation
-Application of Cutting Point Dependent Refinement of Progressive Mesh -

Ryota KITO, Keigo TAKASUGI, Naoki ASAKAWA and Masato OKADA

Reduction of calculation time of tool path generation for complex 3D models is an important issue in development of CAM applications. In this study, cutting point dependent progressive mesh is applied for the models which are represented by meshes. This method maintains history of simplification of the 3D model with hierarchical structure. 3D model can be refined and simplified by using the cutting point dependent progressive mesh. Therefore, the tool path is safely generated by reducing the number of meshes appropriately. Moreover, calculation time can be reduced. In this paper, a tool path is generated by using cutting point dependent progressive mesh, and the calculation time and the result of tool path generation are evaluated. As a result, calculation time of tool path generation was reduced by use of cutting point dependent progressive mesh. And, it is confirmed that the while result of collision check is same before simplification by upping level of detail.

Key words: CAM, tool path generation, collision check, QEM method, view dependent progressive meshes

1. 緒 言

工具姿勢を自由に決定できる多軸制御加工機においてCAMは必須である。一般的に、CAMで3Dモデルを構成する自由曲面はB-SplineやNURBSなどのパラメトリックな幾何表現手法が用いられているが、複雑な曲面になればなるほどその曲面を表現するために必要な階数や制御点数は大きくなり、幾何計算に多くの演算リソースを要することになるため、干渉検出に掛かる計算時間が長くなる。例えば図1は制御点72点で構成された1枚のNURBS曲面に対して、干渉検出を100回繰り返す、その曲面の階数を変化させた時の計算時間を示している（CPU：Intel Core i5-3570、メモリ：8Gbyte、OS：Windows 7 Professional 64bit、CPUのコアは1つ使用）。このようにモデルが複雑になれば、NURBS曲面の階数も高くなり、干渉検出に掛かる時間が長くなってしまう。よって、工具経路生成に多大な計算時間を要することになるため、干渉検出の計算時間の短縮はCAMにおいて非常に重要な課題である。

近年CAMに関するさまざまな研究が行われている。例えば加工中の工具姿勢と姿勢変化の影響を評価し、仕上げ加工による適切な工具経路を生成する研究¹⁾や、NURBS曲面を分割し、曲面ごとに適した加工経路生成方法を行うことで、仕上げ面粗さを満たした上で、実加工時間を短縮する研究²⁾などが挙げられる。このように、実際の加工に関する研究は多いが、工具経路生成の計算時間短縮の研究は少ない。

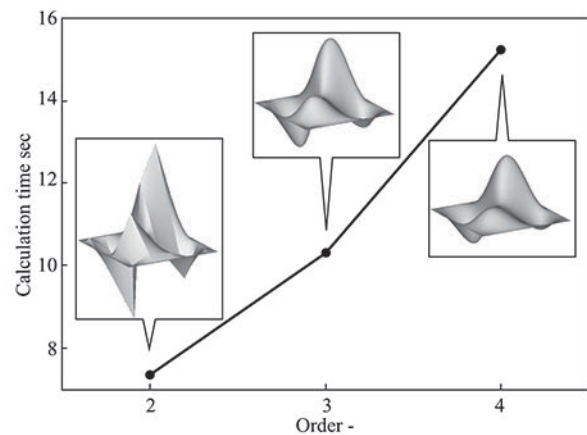


Fig.1 Relation of calculation time and the order of model

計算時間の短縮化を図る研究として、金子らはグラフィックスハードウェアを用いることによって加工可能工具姿勢を高速に決定する方法を提案している³⁾。また森重らは、2次元コンフィギュレーション空間（以下C-Spaceと略す）を用いて各領域の境界線の表現方法を簡略化し、処理を高速化し、曲面補間に基づく工具経路生成を高速にする方法を提案している⁴⁾。

本報ではCAM上の3Dモデルのメッシュ表現に着目した。加工点の生成に関しては、なるべく誤差の少ない幾何計算が望ましく、パラメトリック表現をそのまま用いる必要があるが、干渉検出に対しては、加工点生成ほどの厳密さは必要なく、3Dモデルを小さな三角形の集合であるメッシュで表現する。さらに、メッシュ数をQEM法⁵⁾⁶⁾と視点依存プログレッシブメッシュ表現^{7)~9)}（以下V-PM表現と略す）を応用した加工点依存プログレッシブメッシュ表現を開発することで加

* 原稿受付 平成26年9月2日

掲載決定 平成26年12月15日

** 学生会員 金沢大学大学院（石川県金沢市角間町）

*** 正会員 金沢工業大学（石川県野々市市扇が丘7-1）

**** 正会員 金沢大学（石川県金沢市角間町）

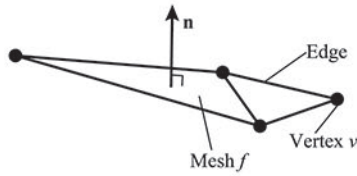


Fig.2 Definition of QEM

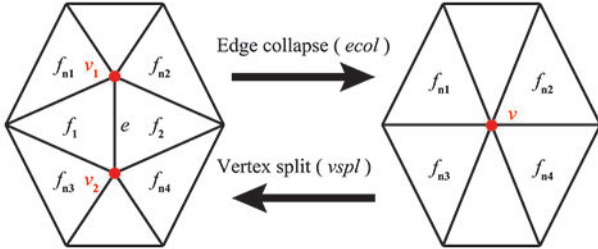


Fig.3 Definition of ecol and vspl

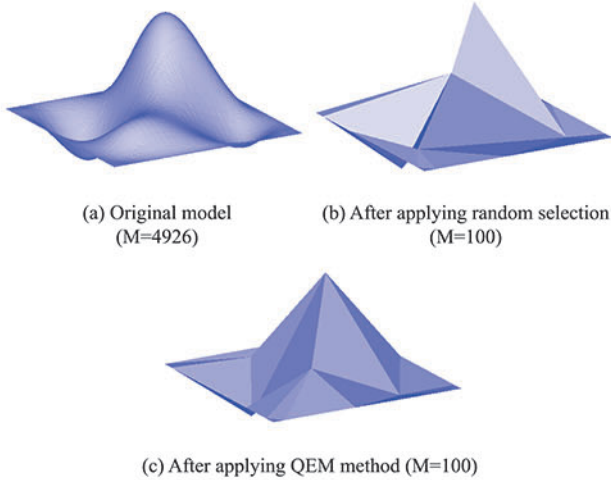


Fig.4 Difference between random and QEM method

工点周辺の幾何誤差を最小限に抑えつつ干渉検出の高速化を図る手法を提案する。

2. Q E M 法

QEM法は3Dモデルの特徴を維持したままメッシュ数を削減する方法であり、例えば、3次元地形データの軽量化¹⁰⁾や医用画像から血管をモデリングし、血流のシミュレーションを行う際に血管のモデルの平滑化を行う方法として用いられている¹¹⁾。

図2のような1つの三角形 f を含む平面と任意の頂点 v の距離は、三角形 f の法線ベクトル \mathbf{n} を用いて、 $|\mathbf{n}^T \mathbf{v} + d|$ で表される。頂点 v 回りの面に対するこの距離の自乗を集めたものがその頂点のQEMとして定義され、

$$Q^v(\mathbf{v}) = \sum_{f \in \mathcal{F}_v} \text{area}(f) |\mathbf{n}_f^T \mathbf{v} + d_f|^2 = \mathbf{v}^T \mathbf{A} \mathbf{v} + 2\mathbf{b}^T \mathbf{v} + c \quad (1)$$

と表現される。ここで、 \mathbf{A} は 3×3 の対称行列、 \mathbf{b} は列ベクトル、 c はスカラーとなっており、 $\text{area}(f)$ は三角形の面積による重み付けである。各頂点 v には $Q^v=(\mathbf{A}, \mathbf{b}, c)$ を構成する10個の値が保存され、メッシュの形状を表現する。 Q^v はメッシュ簡略化の際の評価関数であり、 Q^v が小さいほどその頂点はモデルの形状に影響が少ないことを表している。

実際に3Dモデルの面を消去するには、Edge collapse (以下 $ecol$ と略す)と呼ばれるアルゴリズムを用いる。図3に示

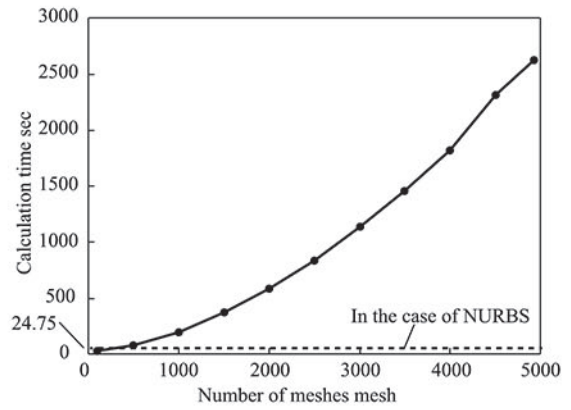


Fig.5 Relation of calculation time and the number of mesh

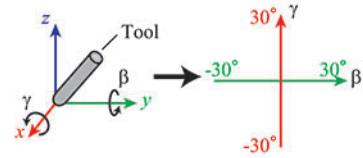
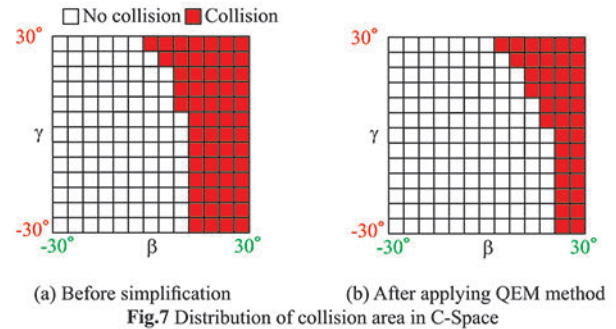


Fig.6 Definition of the C-Space



すように、辺 e を消去し、その辺 e の両端の頂点 v_1, v_2 を結合し頂点 v を生成する。そして、辺 e 以外の頂点 v_1, v_2 を用いている辺の頂点を v に変更する。この消去する辺を決定する方法は、辺の頂点の Q^{v1} と Q^{v2} を加算したものが小さい順に消去を行う。また、この v_1, v_2 を結合してできる頂点 v のQEMは Q^{v1} と Q^{v2} を加算したものを引き継ぐ⁶⁾。

例えば図4(a)に示すメッシュ数 $M=4926$ で表現された3Dモデルを $M=100$ になるまで、ランダムに選んだ面を削除すると、図4(b)のようになり、元の幾何学的特徴を損ねてしまう。しかし、QEM法を同じ3Dモデルに適用し、 $M=100$ まで簡略化すると、図4(c)のように、(b)に比べ、元の幾何学的特徴をよく残していると言える。

QEM法で簡略化することによって、干渉検出計算時間がどのように変化するか調べた。工作物は図4に示したものを使用し、工作物上のある任意の点に工具であるボールエンドミルを移動させ、 ± 30 度の範囲で5度ずつ x, y 軸回りに回転させ、姿勢を変化させた。工作物をNURBS曲面で表現した場合、階数は4階、コントロールポイント数は72である。計算時間は24.75秒かかった。次にメッシュで表現し、メッシュ数を100まで約500ずつQEM法を用いて簡略化し、干渉検出の計算時間とメッシュ数の関係を図5に示す。メッシュ数が多い時ではNURBS曲面の場合より非常に時間が掛かっており、 $M=100$ の時でもようやくNURBS曲面より時間短縮できた。

次に干渉検出の結果を比較した。エンドミルの姿勢を図6に示すように、 x 軸周りの回転を γ 、 y 軸周りの回転を β で表

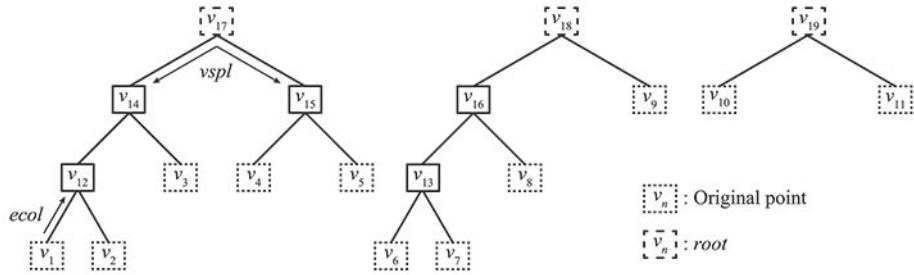


Fig.8 Hierarchical structure of C-PM

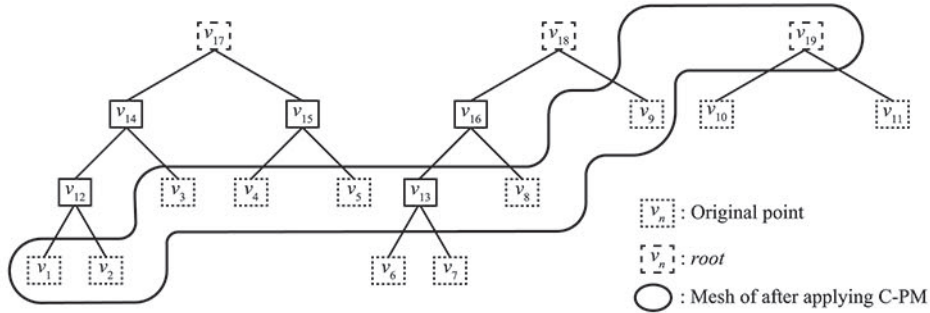


Fig.9 After applying C-PM by refinement of v_{17}

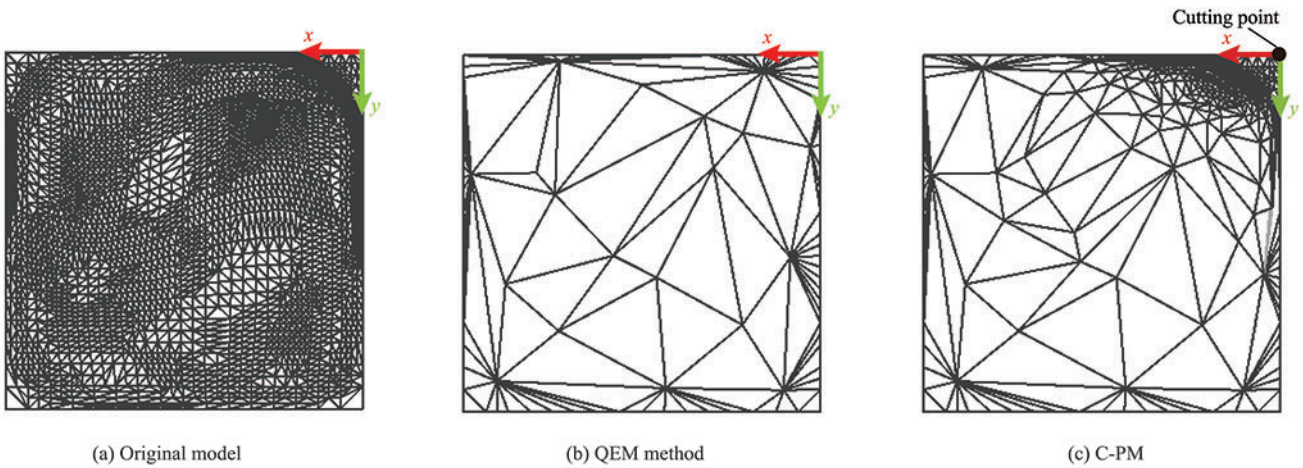


Fig.10 Result of simplification

し、各回転角度時の干渉の有無をプロットした C-Space を作成し、両者を比較する。図 7 に結果を示す。濃い領域が干渉が生じている姿勢である。図 7 (a) と (b) を見てわかるように、簡略化を行ったことで、本来干渉が生じているのに干渉を検出できていない姿勢が存在することがわかる。実際にこの姿勢で加工を行ってしまうと加工が失敗してしまうことから、計算時間を削減しつつ、正確な干渉検出を行わなければならない。

3. プログレッシブメッシュ

3.1 視点依存プログレッシブメッシュ

上記のように、QEM 法で簡略化するだけでは、メッシュ数の減少に伴い、計算時間は減少するが、誤検出が生じてしまう。誤検出を防ぐには、3D モデルをなるべく簡略化しなければよいが、計算時間が掛かってしまう。そこで、工具と工作物間の相対位置を考慮することによって、干渉判定が不要な領域と必要な領域を計算し、干渉判定が必要な領域のみ干渉判定を行い、干渉判定を行うメッシュ数を減らすことが考えられる。しかし、実際にはその領域中に任意のメッシュが含まれているかどうかを判定する計算が必要になり、3D

モデルが複雑になることで、メッシュ数が多くなるとその計算時間が長くなる。また、加工点を変更すると、その加工点での同様の計算をしなくてはならない。よって、3D モデルのメッシュ数が多く加工点が多い場合、全てのメッシュに対して干渉判定対象かどうか判断する計算をするのは非常に時間が掛かると考え、その判断をせずに、尚且つ検出精度を保ちつつ、メッシュ数を削減する方法がないかと考えた。

そこで、本研究では V-PM 表現に着目した。これは一般的にはコンピュータグラフィック分野に用いられる手法であるが、弾塑性解析への応用⁸⁾や、亀裂進展解析、流体解析などにも用いられている¹²⁾。観賞者の視点に近いモデルはメッシュを密にすることで高精細な描画を行い、遠ざかるにつれてメッシュを粗くすることで、レンダリングの処理時間を減らし、滑らかな映像を表現する手法である。V-PM 表現を応用することで、曖昧であるが計算時間を短くしながら、加工点近傍のメッシュは密に、それ以外のメッシュは粗くすることができると考えた。

V-PM 表現は、QEM 法によって *ecol* を行う際に、変更されたデータをメモリバッファに階層構造として保存しておくことで、*ecol* とその逆操作 *Vertex split* (以下 *vspl* と呼ぶ) をバッファ読み出し操作のみで可能にするため、高速なメッシュの

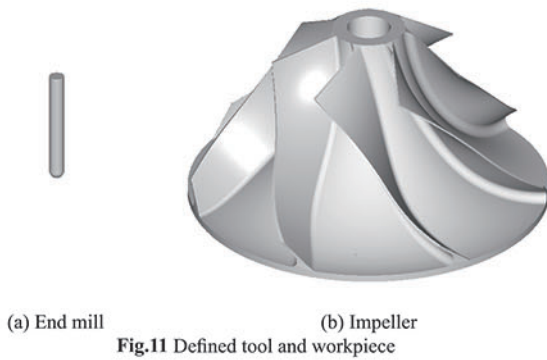


Table 1 Number of meshes and NURBS surfaces of 3D models

	End mill	Impeller
Mesh	590	17078
NURBS	5	65

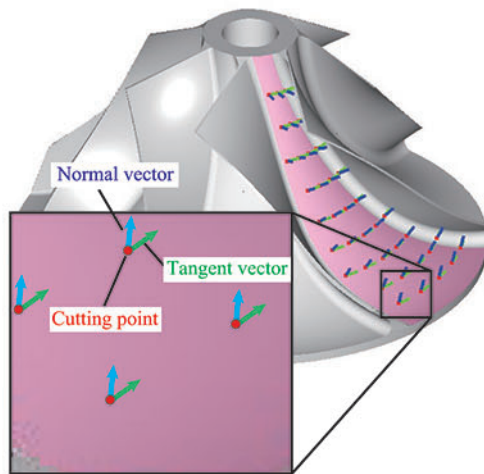


Fig.12 Generation of cutting points, normal vectors, and tangent vectors

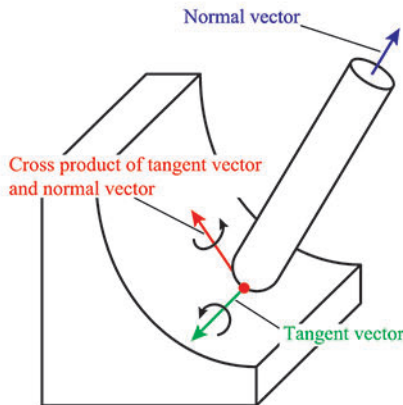


Fig.13 Determination of tool posture

粗密変更を可能にする。ecolは図3に示すように、 $f_{n1} \sim f_{n4}$ が $f_1 \sim f_2$ に隣接している。 $vspl$ は v の周囲に $f_{n1} \sim f_{n4}$ が全て存在していることが条件である。

3.2 加工点依存プログレッシブメッシュ

V-PM表現はもともと観賞者の視点を基準にメッシュの粗密を決定する手法として開発されたアルゴリズムである。本研究では、 $vspl$ の条件や、保存構造を階層構造にしている点はV-PM表現と同様であるが、メッシュの粗密を付けるパラメータを視点から加工点に変更することで、干渉判定の高速性を確保しつつ、その精度の向上を試みる。本手法を加工点依存プログレッシブメッシュ表現(C-PM表現と略す)と呼ぶ

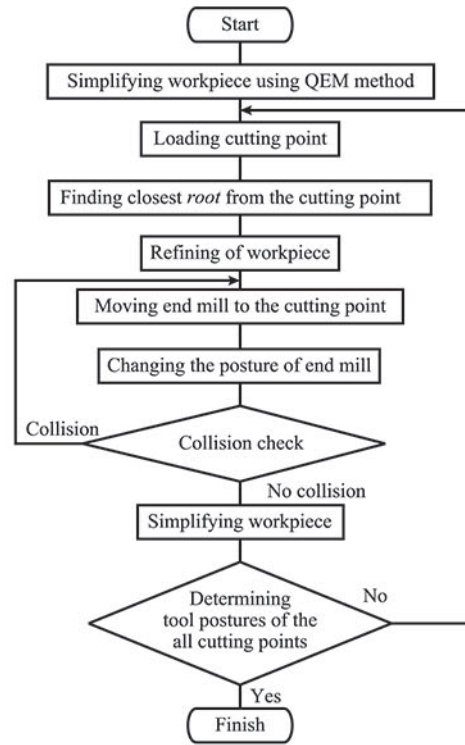


Fig.14 Flowchart of determination of tool postures in C-PM

ぶことにする。以下にC-PM表現の手順を示す。

- 1) 十分な精度を確保したメッシュ表現の3Dモデルを用意する。このオリジナルモデルの点をオリジナルポイントと呼ぶことにする。
- 2) QEM法を用いて簡略化する。この時、変更された点は図8に示すように階層構造として保存される。簡略化後に残った点は階層構造の最上階に位置し、*root*と呼ぶ。
- 3) 加工点の座標から最も近い*root*を探索する。
- 4) 探索した*root*に属するオリジナルポイントを全て使うように詳細化していく。例えば、図8の v_{17} が加工点から最も近い*root*の点だとすると、図9のように詳細化される。

以上の操作によって、加工点近傍のみが詳細なメッシュで表現化された3Dモデルを得ることができる。各メッシュに対して干渉判定対象かどうか判断せずにこの詳細化した3Dモデル全体に対して干渉判定を行えば、精度と速度の両立が可能である。

図10にQEM法及びC-PM表現の適用例を示す。図10(a)は簡略化前のモデルであり、(b)にはQEM法での簡略化後のモデル、(c)にC-PM表現を適用したモデルを示す。加工点近傍は(a)のように詳細なメッシュになっているが、加工点から遠い部分では、(c)のような簡略化されたメッシュになっていることがわかる。簡略化前が $M=4926$ 、簡略化後は $M=128$ 、詳細化後が $M=716$ となっている。また、QEM法による簡略化に掛かる計算時間は12.57秒であったが、詳細化には0.03秒とほとんど時間が掛かっていない。以上の結果から、V-PM表現を応用したC-PM表現を適用すると、メッシュ数を大幅に削減しつつ、加工点近傍は詳細なメッシュにすることができる。

4. 工具経路生成計算時間の検証

4.1 検証条件

工具経路生成を行い、計算時間の比較を行った。比較対象

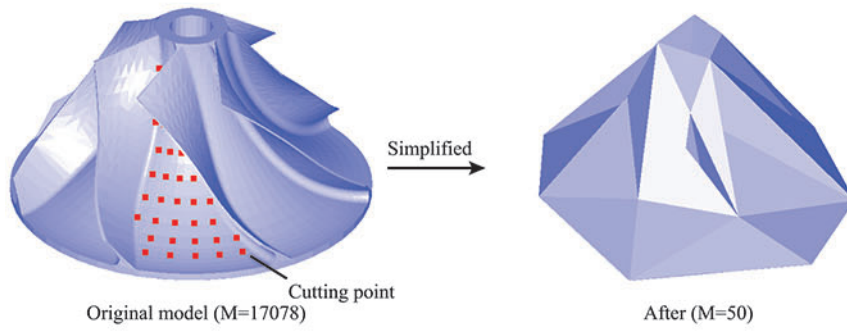


Fig.15 Comparison between original and simplified 3D model

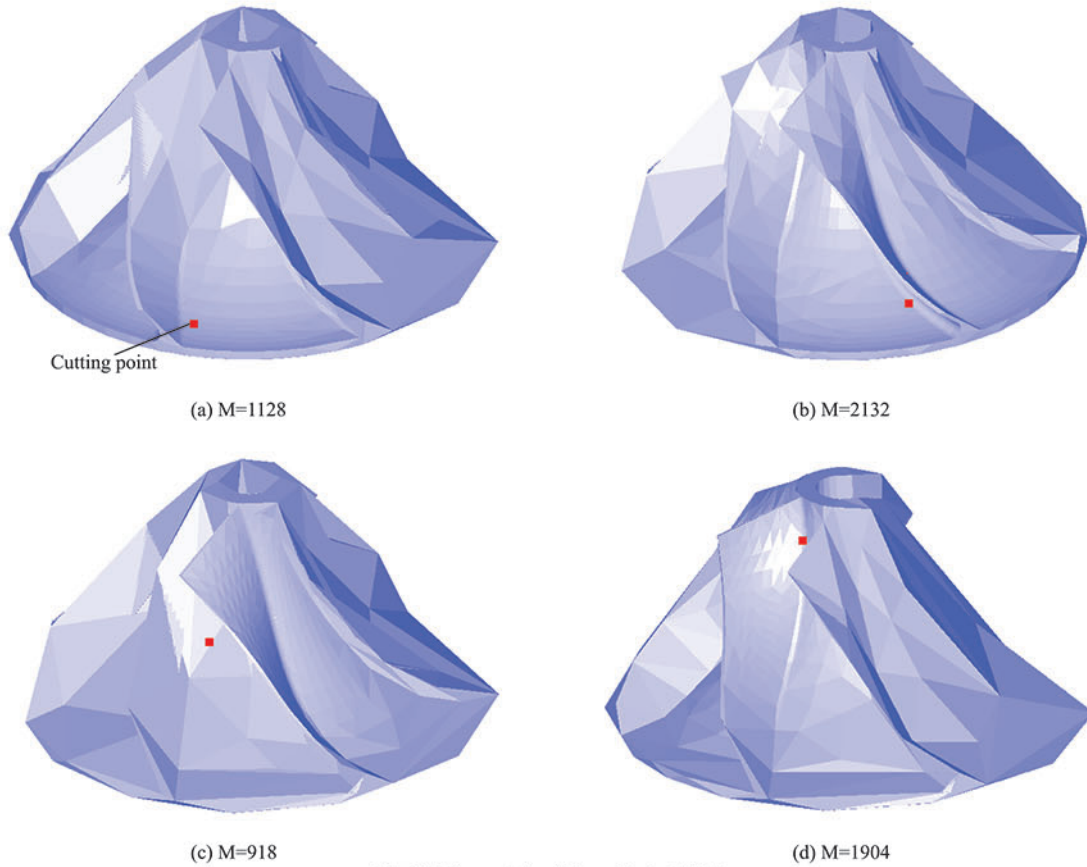


Fig.16 After applying 3D models by C-PM

Table 2 Calculation time of tool path generation

Condition	Calculation time sec
i)	1895.13
ii)	479.38
iii)	180.71

を以下に示す.

- i) 工具, 工作物を NURBS で表現した場合
- ii) 工具, 工作物をメッシュで表現した場合
- iii) 工具, 工作物をメッシュで表現し, 工作物に C-PM 表現を用いた場合

また, 今回用いた工具, 工作物のモデルは, 図 11 に示す $\phi 4$ のボールエンドミルと一般的な形状をしたインペラを用いた. NURBS あるいはメッシュで表現した場合の工具, 工作物の面数を表 1 に示す.

加工点は NURBS 曲面から算出する. また, 工具姿勢決定のため各加工点における法線ベクトルと接線ベクトルを生成する. これは法線ベクトルと接線ベクトルを用いて, 下記に示す方法を用いて工具姿勢を決定するためである. インペラ

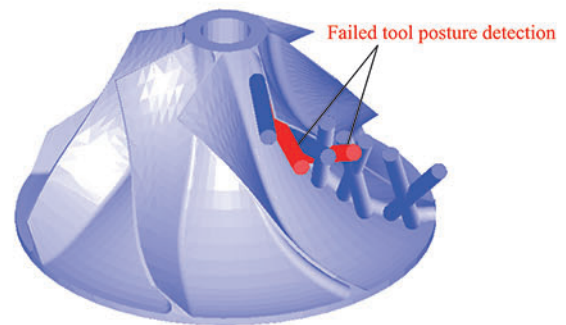


Fig.17 Result of tool path generation

の羽と羽の間の 1 面を選択し, 加工点, 法線ベクトル, 接線ベクトルを算出した結果を図 12 に示す. 今回は 38 点の加工点を生成し, 工具姿勢を決定した.

工具姿勢を決定するため, エンドミルを加工点に移動させ, エンドミル中心軸の姿勢がその加工点の法線方向になるように変化させる. この時のエンドミルの姿勢を初期姿勢と定義する. 図 13 に示すように, この初期姿勢から, 加工点を中

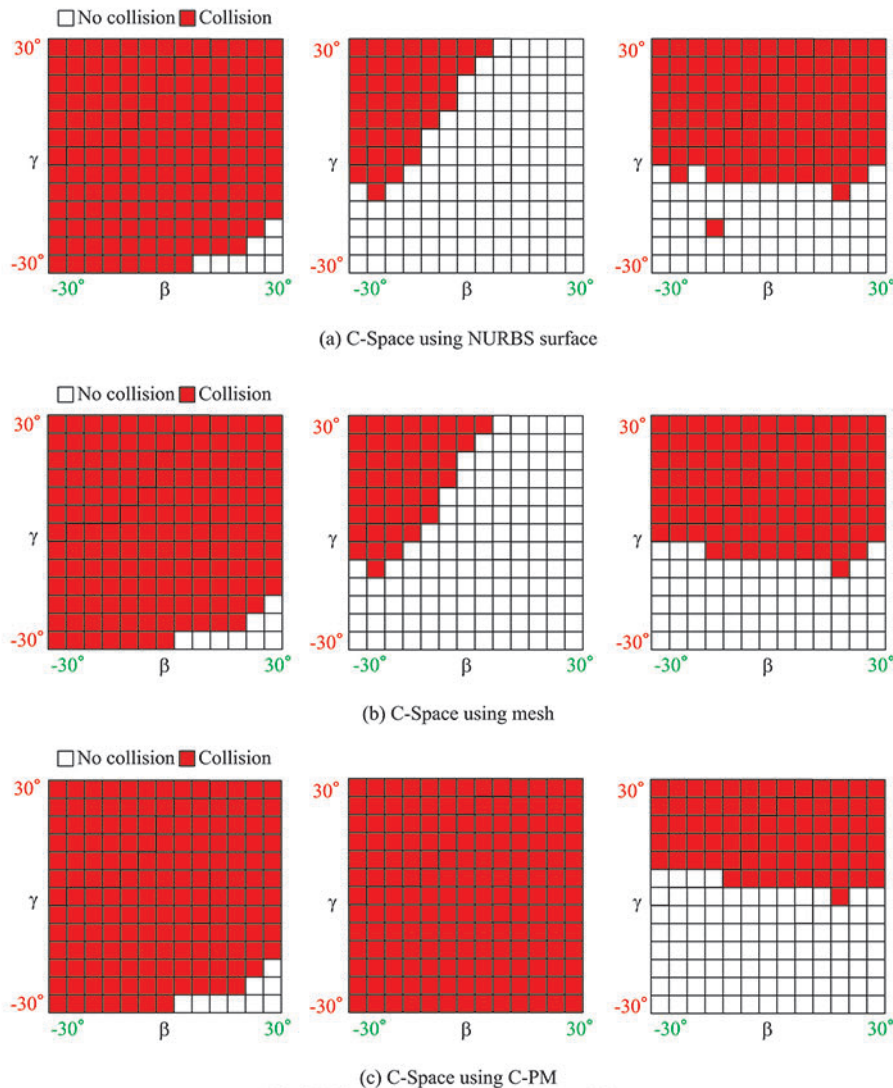


Fig.18 Distribution of collision area at C-Space

心としてエンドミルを± 30度の範囲で5度ずつ接線ベクトルおよび、接線ベクトルと法線ベクトルの外積周りに回転させる。もし、工具と工作物の干渉が生じない姿勢を見つけたら、その加工点においては計算を終了する。全ての加工点での工具姿勢を決定するまでの時間を測定する。また、iii) の場合は、QEM法を用いて工作物を簡略化する。その後加工点を読み出し、その加工点の座標を用いてC-PM表現を適用し、工作物の詳細化を行い、工具姿勢を決定する。工具姿勢が決定したら、工作物をQEM法で簡略化した状態に戻し、次の加工点に移動する。アルゴリズムフローチャートを図14に示す。また、iii) での工作物モデルのメッシュの変化を図15、図16に示す。図15には、QEM法での簡略化前と簡略化後の工作物を示し、図16には、加工点が変わることによって、どのように変化したかを4パターン示す。また、工作物のメッシュ数は平均でM=1518となり、87.4%メッシュ数を削減することができた。

4.2 計算時間比較

i) からiii) それぞれの場合の計算時間を表2に示す。C-PM表現を用いることでi) と比べiii) の場合では90.4%計算時間が減少した。また、ii) の時と比べると62.3%減少した。

いずれの場合もC-PM表現を適用することによって、計算時間を減少させることができたが、工具経路生成結果を比べると、i) とii) では工具姿勢を決定できていた。一方iii)

では加工可能な姿勢が見つからず、工具姿勢が決定できない点が見受けられた。図17にiii) での工具経路生成結果の1列分を表示した結果を示す。青い工具で示されているのは工具姿勢が決定できた加工点、赤い工具で示されているのは工具姿勢が決定できなかった加工点である。これは、C-PM表現を用いることで干渉検出結果がC-PM表現適用前から異なっていることが原因と考えられる。そこで、干渉検出が問題なくできているか調べるため、加工点3点分に対してC-Spaceを作成した。その結果を図18に示す。図18(a)にi) の場合、(b)にii) の場合、(c)にiii) の場合を示す。図18(a)と(b)を比べると、詳細メッシュの場合であってもわずかに干渉検出結果が異なる場合があるが、概ね一致していることがわかる。

しかし、図18(c)の、C-PM表現適用後の結果では、図18(b)と干渉検出結果が著しく異なっている。また、本来干渉している箇所が検出できていない工具姿勢も存在する。よって、C-PM表現を用いて詳細化したモデルの精度は不十分であることがわかった。

4.3 複数点詳細化によるC-PM表現

上記の結果では、工作物の詳細化が足りなく、工具経路生成や干渉検出に不備があった。よって、工作物をより詳細化する必要があると考え、その度合いを詳細化レベルと定義する。上記では、詳細化するrootの点を1点にしているが、この点の数を増やすことが、工作物の詳細化レベルを上げるこ

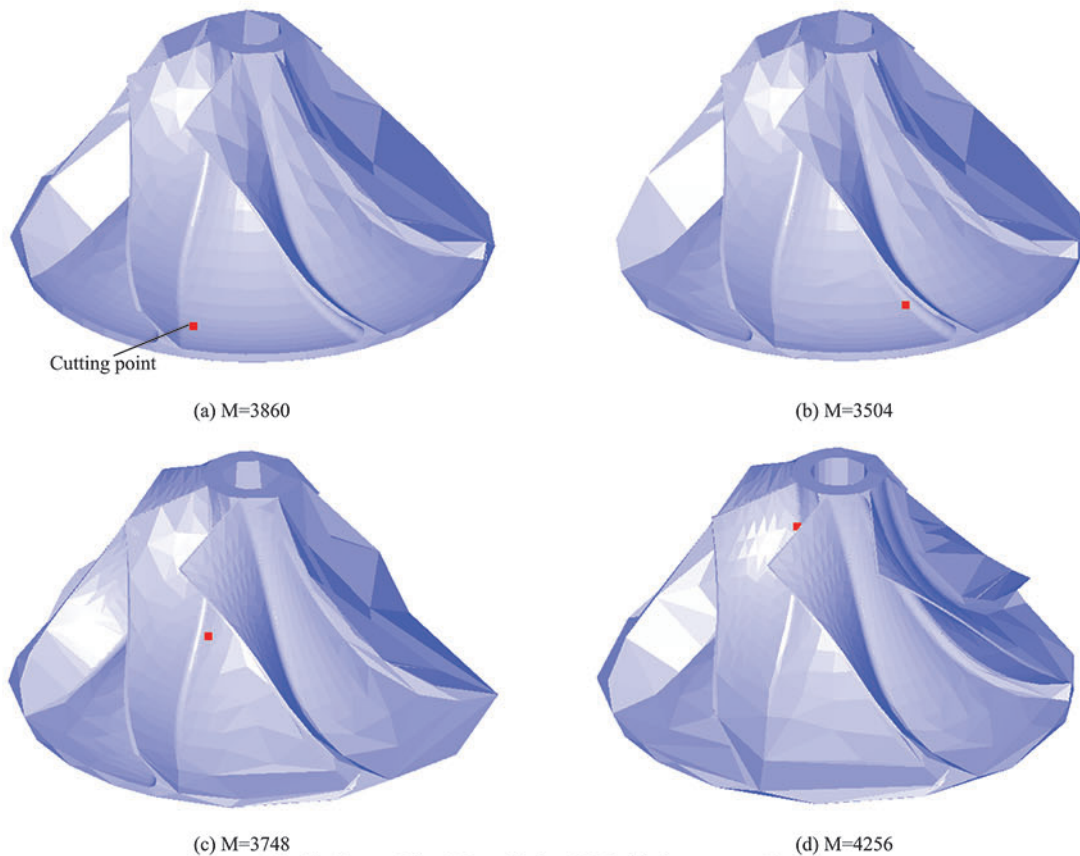


Fig.19 After applying 3D models by C-PM with three roots refinement

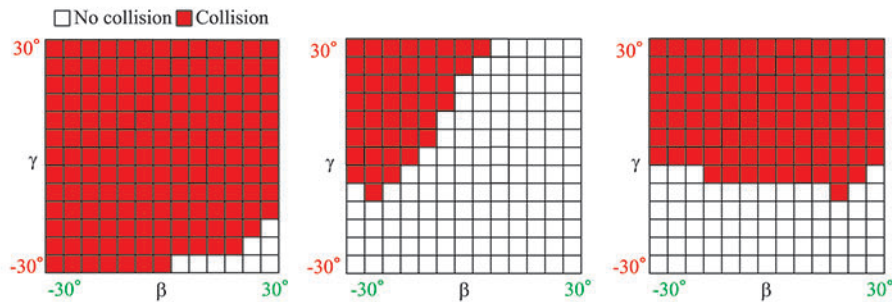


Fig.20 Distribution of collision area at C-Space with three roots refinement

とに相当する。なお、詳細化する *root* の点数を増加させる場合は加工点から近い順に選択することとした。これは工作物を QEM 法で簡略化した際、メッシュ数が $M=50$ で *root* の点数が 20 点と非常に少なく、加工点に近い順に選択するだけで、十分に詳細化できると考えたためである。

詳細化する *root* を 3 点にした場合の工作物のメッシュの変化を図 19 に示す。この時メッシュ数の平均は $M=3849$ となり、詳細化する *root* が 1 点の時より、約 2.5 倍となった。工具経路生成結果は ii) の場合と全く同じになり、計算時間は、206.17 秒になり、詳細化する *root* が 1 点の時より、計算時間は 14.1% 長くなったが、i) の場合より計算時間は、89.1% 短くでき、ii) の場合より 57.0% 短くできた。図 17 と同じ加工点において C-Space を作成した結果を図 20 に示す。干渉検出結果がメッシュの場合と全く同じになった。

以上の結果より、詳細化レベルが低いと不十分であった工具経路生成や干渉検出結果だったが、詳細化レベルを上げることで、工具経路生成時間や工具経路生成と干渉検出において、非常に良い結果が得られた。また、加工点付近の *root* を詳細化したため、大きなオーバーハング形状に対しては加工点から遠い部分で干渉誤差が生じる可能性が残っており、特

殊なモデルによっては *root* 選択の方法の検討が必要であるが、本検証で用いた一般的なインペラ形状の場合では、十分効果的であることを確認した。

5. 結 言

本研究は工具経路生成時の計算時間短縮を目的として、V-PM 表現を応用した C-PM 表現による工作物の簡略化を提案した。また本手法を用いて工具経路生成の計算時間を簡略化前の計算時間と比較を行った。さらに、ある加工点 3 点分で干渉検出結果による C-Space を作成し、簡略化前と比較を行い、以下の結論を得た。

- (1) C-PM 表現を用いることで加工点近傍は詳細で、それ以外は簡略化されたメッシュを生成し、工作物のメッシュ数を的確に削減することができた。
- (2) インペラを工作物として、工具経路生成を行い計算時間を測定し、NURBS 曲面とメッシュで比較を行い、NURBS 曲面で約 83%、メッシュで約 34% 計算時間を短縮することができた。
- (3) 詳細化レベルを上げることで、工具経路生成結果は簡略

化前と同様にし、計算時間を短縮することができた。

- (4) 詳細化レベルが低いと、干渉検出結果に不備があったが、詳細化レベルを上げることで、干渉検出結果がメッシュのときと同じになった。
- (5) C-PM 表現は大きなオーバーハング形状などといった特殊なモデルでは詳細化方法の更なる検討が必要であるが、一般的なモデルでは十分に効果を発揮した。

参 考 文 献

- 1) 引地達哉, 中本圭一, 石田徹, 竹内芳美: 工具姿勢および姿勢変化を考慮した 5 軸制御加工用工具経路の生成, 日本機械学会 2010 年度年次大会講演論文集, (2010), 207
- 2) 穂積昭充, 朱疆, 田中智久, 齋藤義夫: NURBS 曲面の分割による高効率加工パスの生成, 2013 年度精密工学春季大会学術講演会講演論文集 (2014), 1065
- 3) 金子順一, 堀尾健一郎: 5 軸制御切削加工における工具姿勢のグラフィックスハードウェアを用いた高速な決定法 - 被削物のオフセット面と視線群との交差判定を利用した加工可能工具姿勢範囲の導出 -, 精密工学会誌, 72, 8(2006), 1012

- 4) 小畑智晴, 森重功一, 竹内芳美: 2次元 C-Space を利用した 5 軸制御加工のための工具経路生成法 - 複雑形状加工に適用するための処理の高速化と堅固化を重視した改良 -, 精密工学会誌, 70, 2(2004), 287
- 5) M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Computer Graphics (Proc. SIGGRAPH 97)*, 209. ACM Press, New York, 1997.
- 6) 金井崇: 多重解像度メッシュの生成・編集処理技術, 情報処理学会誌, 41, 10(2000), 1108
- 7) Hoppe, H., View-dependent refinement of progressive meshes, *In Computer Graphics, SIGGRAPH'97 Proceedings*, (1997), 189
- 8) 黒川哲郎, 室谷浩平, 矢川元基: 視点依存プログレッシブメッシュとそのアダプティブ弾塑性解析への応用, 日本機械学会論文集, 76, 769(2010), 32
- 9) Hoppe, H., Progressive meshes, *In Computer Graphics, SIGGRAPH'96 Proceedings*, (1996), 99
- 10) 佐藤研一, 今野晃市, 徳山喜政, 曾根順治, 李仕剛: QEM を利用した 3 次元地形データの軽量化, 映像情報メディア学会誌, 56, 12(2002), 2006
- 11) 大島まり: 脳血管障害における医用画像に基づく血流シミュレーション, 日本流体力学学会誌, 21, 2(2002), 122
- 12) Murotani, K. and Yagawa, G., Hierarchical meshing for the adaptive finite elements, *Computational Methods in Applied Sciences*, Vol. 14, (2008), 291, Springer.