パラメトリック曲面上の曲線メッシュ生成アルゴリズムについて

中庭 上総[1,a)]　今井 桂子[1]

概要：本論文では，パラメトリック曲面上に曲線メッシュを生成するアルゴリズムを提案する．パラメトリック曲面に対するメッシュ生成は，CAD や有限要素解析などにおいて重要な手法である．また，どのような性質を持ったメッシュを生成するかは解析の精度に影響を与えることがあり，曲面の曲率などに応じて，指定の方向に引き延ばされた三角形を要素とした非等方性メッシュの方が解析に適している場合がある．既存の手法では，入力として与えられた曲線分を細かい線分で近似して処理するため，曲線分の情報は保持されない．これに対し，提案手法では，曲線分を保護して，一部の三角形の辺が曲線分の一部である曲線メッシュを生成するので，元の曲線を正確に保持することが可能である．これにより，既存手法ではできなかった曲面同士の正確な貼り合わせといった操作が可能になる．本論文では，提案手法の概要と流れを述べる．そして，実際に実装してメッシュを生成し，提案手法の有効性を確認した．

キーワード：曲線メッシュ，パラメトリック曲面，非等方性メッシュ

Curved Mesh Generation on Parametric Surfaces

Kazusa Nakaniwa[1,a)]　Keiko Imai[1]

Abstract: We propose a curved mesh generation algorithm on parametric surfaces. Mesh generation for parametric surfaces is important in CAD or finite element methods. In such applications, the accuracy of the analysis depends on properties of the mesh on the surface. Anisotropic meshes whose elements are elongated according to curvature of the surface is suitable for analysis in some cases. Usually existing methods approximate the curved boundary by many short line segments. On the other hand, by protecting the curved boundary, our method generates curved mesh which is precisely conformed with the input curves, and we are able to patch together surfaces along curved boundary. In this paper, we describe an overview of our method and demonstrate the effectiveness by practically generating curved meshes.

Keywords: Curved mesh, Parametric surface, Anisotropic mesh

1. はじめに

どのような性質を持ったメッシュを生成しておくかは，コンピュータグラフィックスや幾何形状処理，有限要素法といった数値解析手法などの分野において重要である．

メッシュの要素においては，多くの場合，直線分からなる三角形が用いられている．このような三角形のメッシュを生成する手法としては，主要なものとして Delaunay 三角形分割が挙げられる．このアルゴリズムは広く研究されており，様々な三角形メッシュ生成手法のベースになっている．

一方で，要素の辺が高次多項式，つまり曲線で表されるメッシュは，直線の要素からなるメッシュを一般化したものであり，その有用性については [3], [9] などで議論され，実験により示されている．また，曲線境界の準拠（入力の曲線がどれくらい保持されるか）の重要性についても研究

[1] 中央大学大学院
Chuo University, Bunkyo, Tokyo 112–8551, Japan
[†1] 現在，中央大学大学院
Presently with Chuo University
[a)] nakaniwa0206@gmail.com

されている [5]．そして，[6] では，与えられた曲線分を正確に保持して，曲線メッシュを生成するアルゴリズム (Bezier Guarding と呼ばれている) が提案されている（図 1）．この手法は，入力に区分的多項式（Bezier 曲線など）を制約辺として指定することができ，区分的多項式からなる閉曲線を用いることも可能であるが，次元としては平面上に限られている．本研究では，このアルゴリズム中の “Guarding Triangle” という定義を参考にして，メッシュを生成する際に曲線を保護する（詳しくは 2.2 節を参照）．

2 次元平面を少し拡張したものとしてパラメトリック曲面があるが，これに対するメッシュ生成は，CAD やそれに関連するシステムなどにおいて必要な技術である．その中で，有限要素解析においては，正三角形の形に近い三角形で構成される等方性メッシュ（図 2(a)）が求められる．その他に，曲面の曲率などに応じて，指定の方向に引き延ばされた三角形を要素とした非等方性メッシュの方が解析に向いている場合がある（図 2(b)）．また，要素の角度に関しては，形状の悪い（角度が小さいなど）要素が多く存在すると，精密な解析ができないといった影響がでることがある．そのため，品質保証されたメッシュを生成する手法は重要である．

非等方性メッシュを生成する手法としては，Voronoi Refinement[4] がある．これは，角度に関して品質保証（三角形内のどの頂点から計測しても約 20.7°以上になる）されているが，2 次元平面の直線領域にしか適応できない．これを拡張した [7] では，曲線の領域に対応し，角度の品質保証（約 14.4°以上）もされている．その後， [8] では，パラメトリック曲面上に品質保証されたメッシュを生成する手法が提案されている．[7] と [8] では，曲線領域に対応するために，曲線を小さな線分に分割して，Voronoi Refinement を適応するという方法をとっているので，出力されるメッシュは，元の曲線の情報を失ってしまう．

そこで，本研究では，[6] の Guarding Triangle と [8] を参考にして，曲線領域を保持した三角形メッシュをパラメトリック曲面上に生成する手法を提案する．これにより，メッシュ生成後でも曲線の情報が失われないので，曲面同士の正確な貼り合わせといった操作が可能になると考えられる．

本論文の構成は次の通りである．まず，2.2 節, 3 章で，提案手法で用いる定義や用語について述べる．その後，4 章でアルゴリズムの全体的な流れを説明して，5 章で実験結果を示して考察をする．最後に，まとめと今後の課題を述べる．

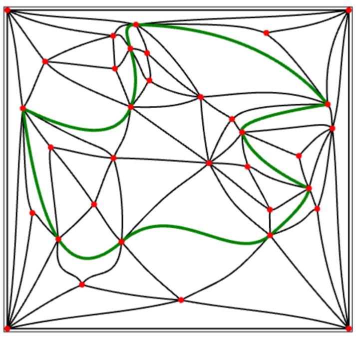
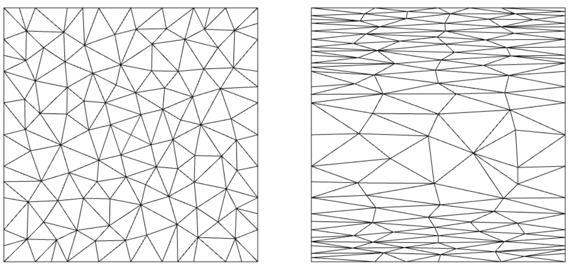


図 1: Bezier Guarding の出力メッシュ



(a) 等方性メッシュ　(b) 非等方性メッシュ

図 2: 等方性と非等方性メッシュ

2. Guarding Triangle

ここでは，入力として与えられる曲線の条件を与え，その後に曲線を保護するための手法である Guarding Triangle の定義と構築方法を説明する．

2.1 入力について

入力は，任意の次数の多項式の曲線 $\boldsymbol{c}_i : [t_{i,0}, t_{i,1}] \to \mathbb{R}^2$ の集合 C であり，一般性を失うことなく $[t_{i,0}, t_{i,1}] = [0,1]$ とおくことができる．曲線 $\boldsymbol{c}_i$ の単位接ベクトルを $\boldsymbol{c}_i^t : [0,1] \to \mathbb{R}^2$ で $\boldsymbol{c}_i^t = \boldsymbol{c}_i' / ||\boldsymbol{c}_i'||$ と書く．入力の曲線 C は以下の 3 つの条件を前提とする (図 3)．

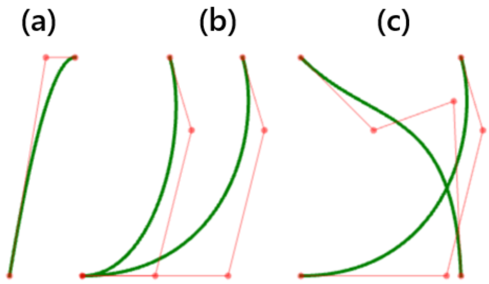


図 3: 入力の条件を満たさない例

(a) ${}^\forall t \in [0,1]$ で $||\boldsymbol{c}_i'(t)|| \neq 0$ となる．
(b) 実数 $s, t \in \{0,1\}$ に対して，$\boldsymbol{c}_i(s) = \boldsymbol{c}_j(t) \Rightarrow \boldsymbol{c}_i^t(s) \neq \pm \boldsymbol{c}_j^t(t)$ ($s = t$ ならば ‘+’，そうでないなら ‘-’)．
(c) ${}^\forall s \in [0,1], t \in (0,1), (i \neq j \vee s \neq t)$ で $\boldsymbol{c}_i(s) \neq \boldsymbol{c}_j(t)$ となる．

条件 (a) は，t の値に関わらず曲線が一点に留まるような曲線は存在しないことを意味し，条件 (b) は，2 つの曲線

の端点が一致する場合において，なす角が 0 度にならないことを意味している．そして，条件 (c) は，1 つの曲線が交差する場合は，端点のみであることを意味している．もしこの条件を満たさないときは，その交点で違反している曲線を分割することで条件を満たすことができる．その場合，交点を誤差無しで求めることは困難だが，幾何的な一貫性を保って有限の精度で丸めて近似する手法が提案されている [1]．

2.2 Guarding Triangle の定義と構築方法

まずは，各曲線を以下の定義を満たすまで（ガード可能になるまで）$t = 1/2$ の点で分割する．分割する方法に関しては，[2] を参考にする．

定義 2.1. $\boldsymbol{p}_i, i \in \{0, ..., n\}$ を次数 n の曲線 $\boldsymbol{c}$ の Bezier 制御点とし，$\boldsymbol{s}_i = \boldsymbol{p}_{i+1} - \boldsymbol{p}_i$ を制御ベクトルとする．このとき，すべての i に対して $\boldsymbol{d}^\mathsf{T}\boldsymbol{s}_i > 0$ を満たす方向 $\boldsymbol{d}$ が存在するとき，かつそのときに限り曲線 $\boldsymbol{c}$ は**ガード可能**であると呼ぶ．

これは，制御ベクトルからなる折れ線の両側において，側の全体が見渡せる（無限遠の点からの）方向が存在することを意味する．例えば，図 4 の曲線はガード可能でないが，このときの制御ベクトルからなる折れ線は，遠方のどの方向から見ても，両側をそれぞれ見渡すことはできない．

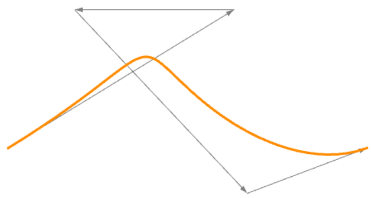


図 4: ガード可能でない曲線の例

そして，次の命題が成り立つ．

命題 2.2. ガード可能でない曲線の分割を繰り返すと，最終的にはすべてがガード可能なサブ曲線になる．

曲線 $\boldsymbol{c}$ の両側において，端点 $\boldsymbol{p}_0, \boldsymbol{p}_n$ を用いてガード点 $\boldsymbol{o}$ を構築する．この点は，曲線 $\boldsymbol{c}$ と直線分 $\overline{\boldsymbol{p}_0\boldsymbol{o}}$ と $\overline{\boldsymbol{p}_n\boldsymbol{o}}$ が三角形領域（一辺は曲線で，二辺は直線）を形成するように選ばれる．

定義 2.3. 曲線 $\boldsymbol{c}$ の，ガード $\boldsymbol{o}$ のある側において，$\boldsymbol{c}, \overline{\boldsymbol{p}_0\boldsymbol{o}}$, そして $\overline{\boldsymbol{p}_n\boldsymbol{o}}$ で形成される三角形を **guarding triangle** と呼ぶ（図 5）．

定義 2.4. ガード可能な曲線 $\boldsymbol{c}$ のすべての Bezier 制御ベクトル $\boldsymbol{s}_i$ に対して，$\boldsymbol{d}^\mathsf{T}\boldsymbol{s}_i > 0$ となる方向 $\boldsymbol{d}$ を曲線の**軸**と呼ぶ．

ある曲線の軸 $\boldsymbol{d}$ に対して，$\boldsymbol{s}^+$ を，$\boldsymbol{d}$ に関して反時計回りで $\boldsymbol{d}^\mathsf{T}\boldsymbol{s}_i$ が最小になる制御ベクトルとし，$\boldsymbol{s}^-$ を時計回りで最小な制御ベクトルとする．$\boldsymbol{s}^+$ 方向で曲線の端点 $\boldsymbol{p}_0$ を通る直点 L^+ と，$\boldsymbol{s}^-$ 方向で曲線の端点 $\boldsymbol{p}_n$ を通る直線 L^- を考える．L^+ と L^- の交点を $\boldsymbol{x}$ とする．もしこれらの直線が平行ならば，その曲線は直線であることを意味する．この場合，$\boldsymbol{x}$ は一点に定まらないので，$\boldsymbol{p}_0$ と $\boldsymbol{p}_n$ の中点とする．そして，$w = ||\boldsymbol{p}_0 - \boldsymbol{p}_n||$ を曲線の幅と定義する．曲線の左側のガード $\boldsymbol{o}_l$ を $\boldsymbol{o}_l = \boldsymbol{x} + \mu(w^2/\widehat{w})\boldsymbol{n}$ とする．ここで，$\boldsymbol{n}$ は $\boldsymbol{d}$ に関して反時計回りに垂直な単位ベクトルであり，$\mu > 0$ はパラメータである．正規化係数 $\widehat{w}$ は，曲線の初期の幅を意味し，曲線を分割しても一定である．μ に関しては，小さな値を設定すると曲線をタイトに囲う三角形を形成し，4 章の曲線の分割がより早く終了するが，得られるメッシュの要素の歪みが大きくなる可能性があるので，トレードオフの関係がある．具体的な値としては，[6] の実験の結果から $\mu = 10^{-2}$ をとっているので，本研究においてもそちらを採用する．

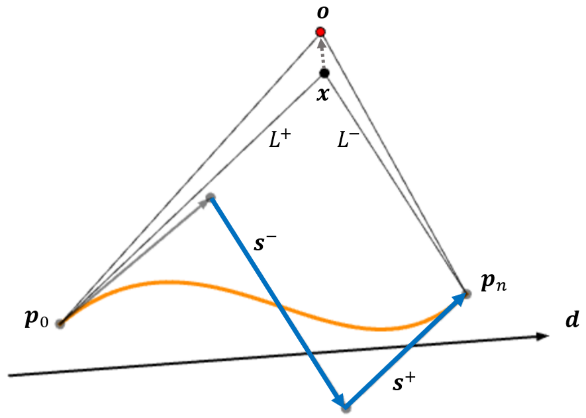


図 5: Guarding triangle の構築

同様の方法で，曲線の右側におけるガード $\boldsymbol{o}_r$ も，曲線を逆さにして考えることで得ることができる．上記の方法で，それぞれの曲線に対して 2 つの guarding triangle を定義することができる．

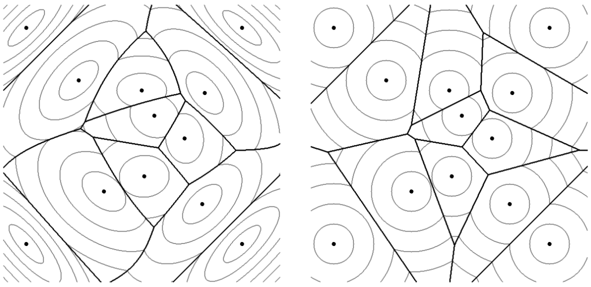
定義 2.5. 曲線 $\boldsymbol{c}$ の端点 $\boldsymbol{p}_0, \boldsymbol{p}_n$ とガード $\boldsymbol{o}_l, \boldsymbol{o}_r$ で形成される四辺形を envelope $E(\boldsymbol{c})$ と呼び，$||\boldsymbol{o}_l - \boldsymbol{o}_r||$ を envelope $E(\boldsymbol{c})$ の大きさと呼ぶ．

上記から分かる通り，曲線 $\boldsymbol{c}$ は $E(\boldsymbol{c})$ の中に含まれ，曲線の端点のみが $E(\boldsymbol{c})$ 上にある．

定義 2.4 によれば，軸 $\boldsymbol{d}$ は無数に存在する．そこで，$\boldsymbol{s}^+$ と $\boldsymbol{s}^-$ を二等分する向き，すなわち $\boldsymbol{d} = \frac{\boldsymbol{s}^+}{||\boldsymbol{s}^+||} + \frac{\boldsymbol{s}^-}{||\boldsymbol{s}^-||}$ とする．定義 2.1 から，ガード可能な曲線 $\boldsymbol{c}$ には，$\boldsymbol{d}^\mathsf{T}\boldsymbol{s}^+ > 0$ かつ $\boldsymbol{d}^\mathsf{T}\boldsymbol{s}^- > 0$ となる方向 $\boldsymbol{d}$ が存在するので，このように定めても問題はない．

3. パラメトリック曲面に対する非等方性メッシュ生成

本節では，パラメトリック曲面に対する非等方性メッシュを生成する手法 [8] について説明する．まず，入力と



(a) 非等方性 Voronoi 図　(b) 等方性 Voronoi 図

図 6: 非等方性と等方性 Voronoi 図

して与えられるテンソル場とパラメトリック曲面について簡単に説明し，その後に非等方性メッシュを生成する流れを述べる．

3.1 テンソル場

テンソル場は，メッシュを生成したい領域 Ω に対して，どのようなメッシュを生成するかを指定するために与えられる．領域 Ω 内のどの点 $\boldsymbol{p} \in \Omega$ に対しても計量テンソル $M_{\boldsymbol{p}}$ が定義される．例えば，2 点間の距離は，Euclid 空間のような等方性のときとは異なる．点 $\boldsymbol{p}$ から計測した 2 点 $\boldsymbol{q}_1, \boldsymbol{q}_2 \in \Omega$ 間の距離は

$$d_{\boldsymbol{p}}(\boldsymbol{q}_1, \boldsymbol{q}_2) = \sqrt{(\boldsymbol{q}_1 - \boldsymbol{q}_2)^\top M_{\boldsymbol{p}}(\boldsymbol{q}_1 - \boldsymbol{q}_2)}$$

となる．角度に関しては，点 $\boldsymbol{p}$ から計測した角度 $\theta = \angle \boldsymbol{q}_1\boldsymbol{q}_2\boldsymbol{q}_3$ は

$$\theta = \arccos \frac{(\boldsymbol{q}_1 - \boldsymbol{q}_2)^\top M_{\boldsymbol{p}}(\boldsymbol{q}_3 - \boldsymbol{q}_2)}{d_{\boldsymbol{p}}(\boldsymbol{q}_1, \boldsymbol{q}_2) d_{\boldsymbol{p}}(\boldsymbol{q}_3, \boldsymbol{q}_2)}$$

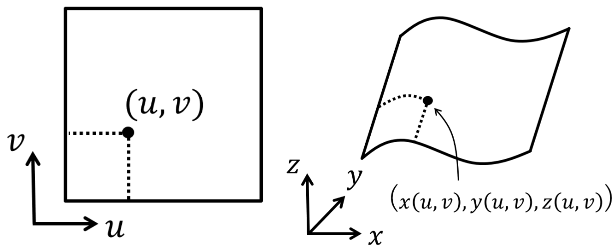
となる．この距離計量における Voronoi 図は図 6(a) のようになる．等方性 Voronoi 図（図 6(b)）は双対を取ると三角形分割が得られるが，非等方性の場合は一般に三角形分割にはならない（裏返った三角形が生じる可能性がある）が，ここで用いる Voronoi Refinement[4] は，双対が三角形分割となるように非等方性 Voronoi 図を生成する．

3.2 パラメトリック曲面

パラメトリック曲面 (またはパラメトリック曲面パッチ) とは，3 次元空間に存在する曲面を，2 つのパラメータ u, v $(0 \leq u, v \leq 1)$ を用いて表した曲面のことである．3 次元空間における曲面上の点の座標 (x, y, z) は，パラメータ u, v で決まり，曲面の関数 $P : \mathbb{R}^2 \to \mathbb{R}^3$ は

$$P(u, v) = (x(u, v), y(u, v), z(u, v))$$

と表せる．ここで，曲面の存在する 3 次元空間を実空間と呼び，パラメータ u, v からなる 2 次元空間をパラメータ空間と呼ぶ（図 7）．そして，トリム曲面とは，パラメトリック曲面の一部分を切り出したものである．一部分を切り出すには，通常，パラメータ空間上にトリム領域を指定する．



(a) パラメータ空間　(b) 実空間

図 7: パラメトリック曲面

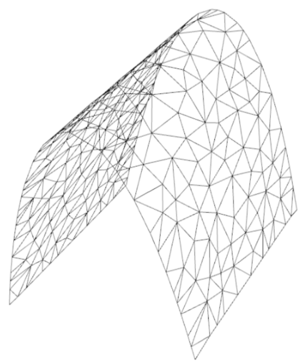


図 8: パラメトリック曲面上の非等方性メッシュ

3.3 非等方性メッシュ生成の流れ

入力としてパラメトリック曲面 (またはトリム曲面) と実空間上のテンソル場が与えられたら，まず，テンソル場をパラメータ空間上のテンソル場に変換する．そして，そのテンソル場を利用して，パラメータ空間上に Voronoi Refinement を適応して非等方性メッシュを生成する．最後に，得られたメッシュを実空間上に射影する．図 8 は，実際にパラメトリック曲面上に生成した非等方性メッシュである．

4. 提案手法の流れ

本研究では，パラメトリック曲面上に曲線領域の非等方性メッシュを生成する手法を提案する．アルゴリズムの流れは次の通りである（図 9）．

Step 1: 入力として，パラメトリック曲面，実空間上のテンソル場，そしてトリム領域を与える．

本研究では，パラメトリック曲面として Bezier 曲面を，トリム領域として Bezier 曲線を用いることとする（図 9(a)）． パラメータ空間上で非等方性メッシュを生成するために，実空間に与えられたテンソル場をパラメータ空間上のテンソル場に変換する．

Step 2: トリム領域の各曲線に対して guarding triangle を構築する．

2.2 節に従って， 各曲線の両側に guarding triangle を構築する（図 9(b)）．その後，各曲線のペア $\boldsymbol{c}_i, \boldsymbol{c}_j$ の，曲線の端点 $\boldsymbol{p}_0, \boldsymbol{p}_n$ を除く envelope $E(\boldsymbol{c}_i), E(\boldsymbol{c}_j)$ が交

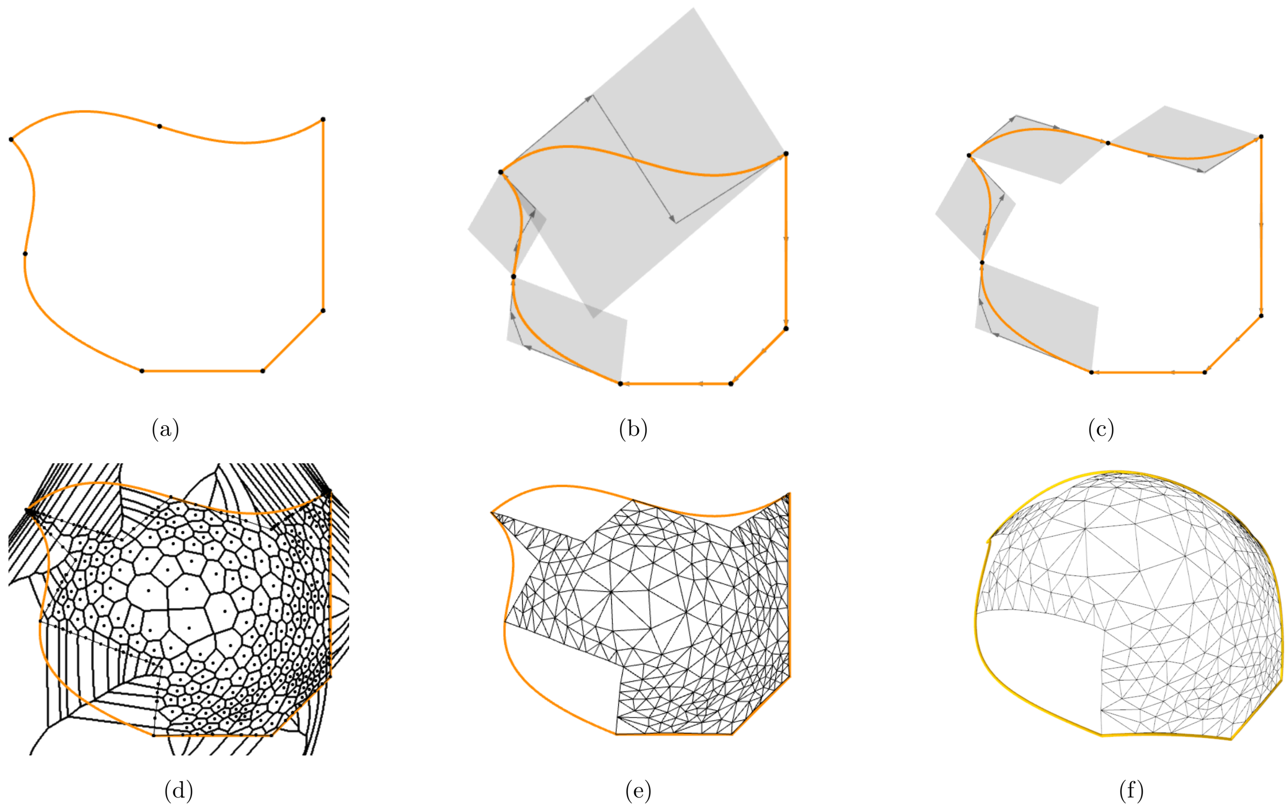


図 9: 提案手法の流れ

差している場合は，envelope の大きさを比較して，大きい方の曲線を分割する．これをどの曲線のペアも交差しなくなるまで繰り返す（図 9(c)）．分割については，2.2 節と同様に二等分割する．

Step 3: トリム領域内部の guarding triangle からなる領域内に，品質保証付き非等方性メッシュを生成する．
各曲線におけるトリム領域内部の guarding triangle は，直線分からなる多角形領域を形成する．その領域内に 3.3 節の手法を用いて非等方性 Voronoi 図を生成する（図 9(d)）．そして，双対を取ることで角度保証付きの非等方性メッシュを得る（図 9(e)）．

Step 4: 得られた非等方性メッシュをパラメトリック曲面上に射影する．
射影して得られる曲面上のメッシュをアルゴリズムの出力とする（図 9(f)）．

5. 実験と考察

ここでは，実際にパラメトリック曲面上に非等方性メッシュを生成した結果と角度の分布等をいくつか示す．具体的には，最初に 1 つのメッシュを生成した結果を示し，次に 2 つのメッシュを生成し，それらを貼り合わせる実験の結果を示す．そして，それらに対する考察を述べる．

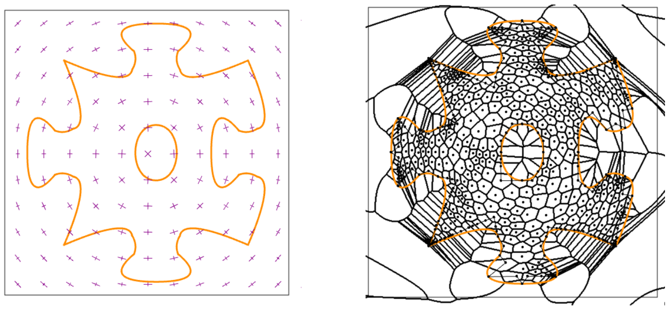
5.1 実験結果

まず図 10 は，1 つのトリム領域内にメッシュを生成しない領域を与えて，提案手法を適応したときの結果である．そして，図 11 と表 1 はそのメッシュの角度に関するデータである．ここでの角度とは，三角形で近似された曲面内の任意の点から計測した角度のことを言う（実際には，数値的に求めるには三角形内にサンプル点を配置して計測する）．

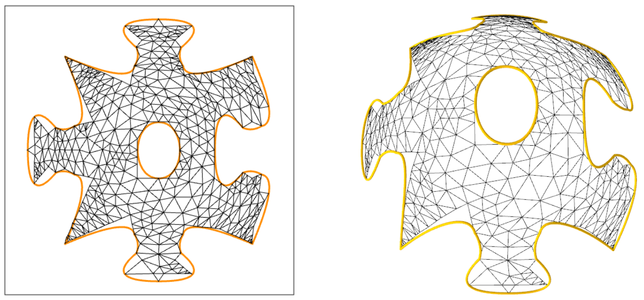
次に図 12 は，パラメータ空間上に 2 つのトリム領域を別々に与えてメッシュを生成し，それらをパラメトリック曲面上で貼り合わせた結果である．そして，図 13 と表 2 はそのメッシュの角度に関するデータである．

5.2 考察

まずメッシュの角度に関しては，表 1，2 の最小角度から分かる通り，どちらも 14.4° 以上になっている．したがって，参考にした手法 [7] の理論的な最小角である $\arcsin(1/4) \fallingdotseq 14.4°$ より大きいことが実際に確認できた．また，相対度数においては，正三角形の内角である 60° に近い値が多く，平均と中央値も同様の結果になった．また，図 12 の結果では，どちらのメッシュも境界は曲線分からなる三角形で構成されているので，入力で与えられた曲線を保持している．そして，guarding triangle は曲線分の両側に構築することができるので，別々にメッシュを生成したとしても，境界を保持することが可能であり，メッシュ生成後のパラメトリック曲面上でも正確に貼り合わせることが可能であることが確認できた．



(a) 曲線とテンソル場　(b) 非等方性 Voronoi 図



(c) 非等方性メッシュ　(d) 実空間に射影

図 10: 複数のトリム領域に対するメッシュ生成

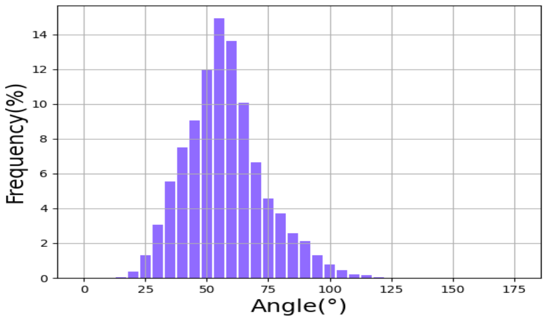


図 11: 図 10 のメッシュの相対度数分布

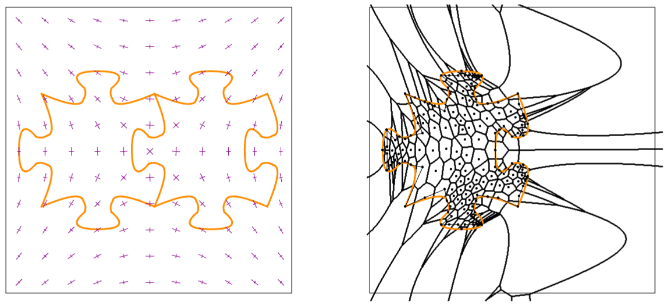
表 1: 図 10 のメッシュの具体的な結果

	角度（°）
最小	15.09
最大	126.21
平均	59.81
中央値	58.74

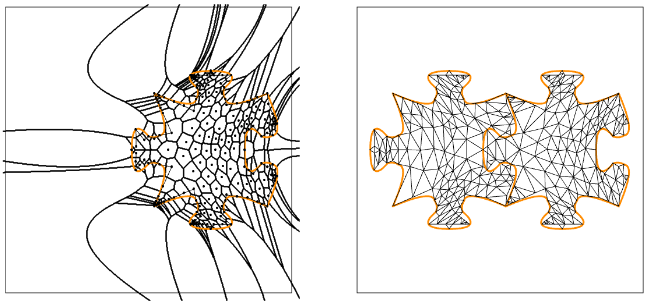
6. おわりに

本研究では，パラメトリック曲面上に曲線三角形を含む非等方性メッシュを生成する手法を提案した．この手法は，[8] のパラメトリック曲面に対するメッシュ生成において課題であった境界の曲線分の保持を，[6] の guarding triangle という曲線をガードするアイデアを参考にして改良することが出来たといえる．これにより，曲面上でのメッシュ同士の正確な貼り合わせなどの操作が可能になると考えられる．

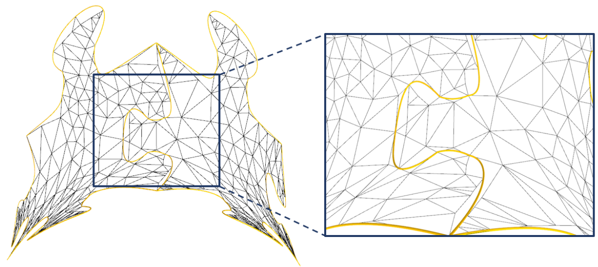
今後の課題としては主に 2 つ挙げられる．



(a) 曲線とテンソル場　(b) 非等方性 Voronoi 図 (左)



(c) 非等方性 Voronoi 図 (右)　(d) 非等方性メッシュ



(e) 実空間に射影

図 12: 曲線メッシュ同士の貼り合わせ

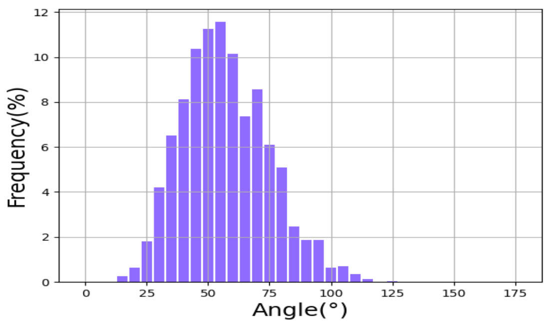


図 13: 図 12 のメッシュの相対度数分布

表 2: 図 12 のメッシュの具体的な結果

	角度（°）
最小	15.04
最大	121.67
平均	59.56
中央値	58.08

1 つ目は出力されるメッシュは曲線三角形（境界上）と直線三角形の 2 種類で構成されるが，それらの大きさが不

揃いな傾向があることである．曲線三角形は，曲線を分割し，guarding triangle を再構築することでより小さい大きさにすることができるが，どれくらい分割すればよいかは，隣接するメッシュ内部の非等方性の領域を考慮する必要があるため，現状では基準を定めるのが難しい．また，曲線を正確に保持したい場合は，曲線の分割の回数が多くなったときの計算コストも課題となる．

2 つ目は，非等方性メッシュを生成する領域である多角形は，内角が鋭角であることが多い傾向があることである．鋭角な箇所が多いと，非等方性メッシュの三角形の角度も全体的に悪くなってしまうので，guarding triangle で形成される多角形の形状についての改善が挙げられる．

謝辞　本研究の一部は，科学研究費補助金 20K116822 の助成を受けたものである．

参考文献

[1] Arno Eigenwillig, Lutz Kettner, and Nicola Wolpert. “Snap rounding of Bezier curves,” In *Proc. Symposium on Computational Geometry.* ACM, pp.158-167, 2007.

[2] Gerald Farin, “Curves and Surfaces for CAGD: A Practical Guide (5th ed.),” Morgan Kaufmann Publishers Inc., 2002.

[3] Xifeng Gao, Yixin Hu, Alec Jacobson, Daniele Panozzo, Teseo Schneider, Qingnan Zhou, and Denis Zorin, “TriWild: Robust Triangulation with Curve Constraints,” *ACM Trans. Graph.*, Vol.38, No.4, 2019.

[4] Francois Labelle, and Jonathan Richard Shewchuk, “Anisotropic Voronoi Diagrams and Guaranteed-Quality Anisotropic Mesh Generation,” *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, pp.191-200, 2003.

[5] Xiaojuan Luo, Jean-Francois Remacle, Mark S Shephard, “The influence of geometric approximation on the accuracy of high order methods,” *Rensselaer SCOREC*, report 1, 2001.

[6] Marcel Campen, and Manish Mandad, “Bezier Guarding: Precise Higher-Order Meshing of Curved 2D Domain,” *ACM Trans. Graph.*, Vol.39, No.4, 2020.

[7] Keiko Imai, and Yusuke Yokosuka, “Guaranteed-Quality Anisotropic Mesh Generation for Domains with Curves,” *The European Workshop on Computational Geometry*, pp.125-128, 2006.

[8] 横須賀 佑介, “パラメトリック曲面に対する品質保証付きメッシュ生成,” 中央大学理工学研究科情報工学専攻修士論文，2005.

[9] Milos Zlamal, “Curved Elements in the Finite Element Method. I,” *SIAM Journal on Numerical Analysis*, Vol.10, No.1, 1973.