Generation of Geodetic Lines and Duplication of Triangulated Convex Surfaces

Anna von Pestalozza
Chair of Short-Time Dynamics
University of the Federal Armed Forces
Hamburg, Germany
pestalozza@hsu-hh.de

Arash Ramezani
Chair of Short-Time Dynamics
University of the Federal Armed Forces
Hamburg, Germany
ramezani@hsu-hh.de

Abstract*—In the following paper, research about geodetic lines as well as about surface duplication is presented. The calculation of geodetic lines plays an important role in many applications, such as the minimisation of material in manufacturing processes. Many manufacturing steps, such as cutting or attaching layers on curved surfaces, suffer from loss of material. In order to minimise wastage of material, geodetic lines can be employed to find a cutting pattern for the given material with minimal distortion. This paper presents an automatable algorithm that numerically calculates geodetic lines on any given surface. The result is evaluated with a practical example by comparing the numerical result and the analytical solution. The creation of multiple layers serves the purpose to reinforce a given structure to increase its stability, which is commonly done in manufacturing processes. This paper presents an algorithm, which calculates the coordinates of multiple attached layers with any given thickness of layers. Furthermore, the point of maximum curvature is determined.

Keywords-Geodetic Line; Surface Analysis; Surface Orientation; Surface Curvature.

I. INTRODUCTION

This paper is based on the assumptions that the material of the given surface is of finite thickness and of low elasticity, which leads to the necessity of minimising loss of material as well as to the necessity to calculate size and coordinates of further layers. The project in which the presented research is embedded aims particularly at protection gear such as helmets and vests. It is therefore presumed that the surfaces which have to be investigated are mainly convex. Moreover, it is assumed that the given triangulation is so fine that on a convex surface the scalar products of the normal vectors of adjacent triangles are always positive. For the implementation of the presented algorithms the software tool Matlab was used for implementation as well as for evaluation.

A general overview over the research about geodetic lines is given in [1], which will be described more in detail in the following paper. The starting point of the research on the geodetic lines is the approach given in [2] for finding geodetic lines between two points. The main idea is to successively calculate distances from the starting point which is improved by the fast marching method. In [1], an algorithm for extracting the geodetic line as well as for further improving it is derived, which will be repeated in the following as well as deepen.

For cutting a curved surface either sectional planes or geodetic lines can be used. The graphical approximation of flattened material stripes of an originally curved surface having been cut by the procedures of applying sectional planes and calculating geodetic lines clearly show that the cutting with the geodetic lines provides straight edges when flattened whereas the sectional planes result in curved edges which leads to a higher amount of material loss. However, sectional planes are much easier to apply and less time consuming than the analytical calculation of geodetic lines which is not even possible in many cases. Thus, this paper aims to provide an algorithm which approximates analytical geodetic lines on any given surface [1].

Using geodetic lines, a cutting pattern can be derived which is not explicitly explained in this paper. However, it is not only one layer which is treated when producing objects so that several layers have to be taken into account. This paper presents an approach how to duplicate layers. For this purpose, finite thinness of the single layers is assumed which is in practice always true so that the overall area of a single layers slightly increases on a convex structure proportional to the number of layers (assuming that the layers are applied on the outer/convex face of the given surface). This shift of coordinates is addressed in this paper as well as the analysis of the orientation of the surface. As a result, an algorithm is presented which uniformly applies layers of any given thickness and size to the *outside* of any given convex surface which will be derived, e.g., which thickens a surface without changing its shape. The algorithm can be easily adapted to shift the coordinates to the inner face if needed.

The paper is divided in nine sections. After the introduction, the calculation of the shortest distance on a triangulated mesh is shown in Section II followed by Section III about the extraction of the geodetic line. In Section IV a straightening algorithm for improvement of the geodetic line is presented. In Section V exemplary results of the applied algorithm which is derived in Section II-IV are evaluated. Section VI explains the basic considerations and concepts of the creation of multiple layers and challenges which have to be faced when duplicating layers. The duplication of layers is then described in Section VII. In

Section VIII the point of maximum curvature. Its relevance for research is shortly paraphrased in Section IX which is a concluding paragraph and an outlook to future work on this project at the end of this paper.

II. Calculation of Shortest Distances on a Triangulated Mesh

In this paper, the procedure of the Fast Marching Method (FMM) is used [4] for calculating the shortest distances on a triangulated mesh. Basically, this method approximates the distances of all points surrounding the starting point successively by a wave front until it reaches the given ending point. For the following procedure it is assumed that starting and ending point of the geodetic line which is to be approximated are given.

A. Procedure

In the FMM, the vertices of all triangles in the mesh are divided into several groups which are sets of vertices.

1) Fixed vertex set (FVS): contains initially only the starting points; vertices which are points of the shortest distance are added in the procedure.

2) Close vertex set (CVS): contains initially no vertices; vertices which are close to the point that is investigated in the current iteration of the loop are added.

3) Unprocessed vertex set (UVS): contains all vertices of the mesh that are not contained in FVS.

Two situations can be distinguished: Only one starting point is given and more than one starting point is given. If there is only one starting point, the distances T_i of its direct neighbours have to be calculated and the neighbours are added to the CVS. If there is more than one starting point, the points a_0, a_1, and a_2 which are part of a triangle of the mesh containing exactly two points in FVS have to be determined. After computing their distances T_0, T_1, and T_2 to the starting value, the points a_0, a_1, and a_2 are added to CVS. After these initial steps, the following loop starts:

- The point a_i, i = 0, 1, 2 with the shortest distance T_i to the starting value is moved to FVS and is now the point of origin for further investigations. This point is called trial.
- The distances T_i of all points in UVS∪CVS which are adjacent to triangles containing trial and a point in FVS are computed and moved to CVS.

In each iteration, one point is added to FVS and its neighbours are added to CVS. The algorithm terminates when FVS contains every vertex which is part of a line resulting in the shortest distance from starting to ending point.

B. Calculation of Distance T

For calculating the distance T the method presented in [3] is used. It requires that one point, P_1, of known distance T_1 is the origin and that another point, P_2, of known distance T_2 is on the x-axis.

1) Procedure: The distance T_3 of the third point P_3 is calculated in terms of T_1, T_2 and the connecting vectors v_i with $v_i = P_i - P_1$, in particular $(v_3)_x$ and $(v_3)_y$, i.e., the projections of v_3 onto the new basis vectors, which are calculated as follows: To change the default, adjust the template as follows:

1. One point is set as the origin (P_1):

$$v_i = P_i - P_1$$

2. The coordinate system is transformed, where:

$$e_x = \frac{(p_2 - p_1)}{|p_2 - p_1|} = \frac{(v_2)}{|v_2|}$$

$$e_y = \frac{(v_3 - e_x \cdot (e_x \cdot v_3))}{\left|(v_3 - e_x \cdot (e_x \cdot v_{3)})\right|} = \frac{(v_3 \cdot |v_2|^2 - v_2 \cdot (v_2 \cdot v_3))}{|(v_3 \cdot |v_2|^2 - v_2 \cdot (v_2 \cdot v_3)|}$$

3. The distance of v_3 to the new coordinate system is computed where O_x is the x-coordinate at which the origin of the new coordinate is located and O_y is the relative y-coordinate:

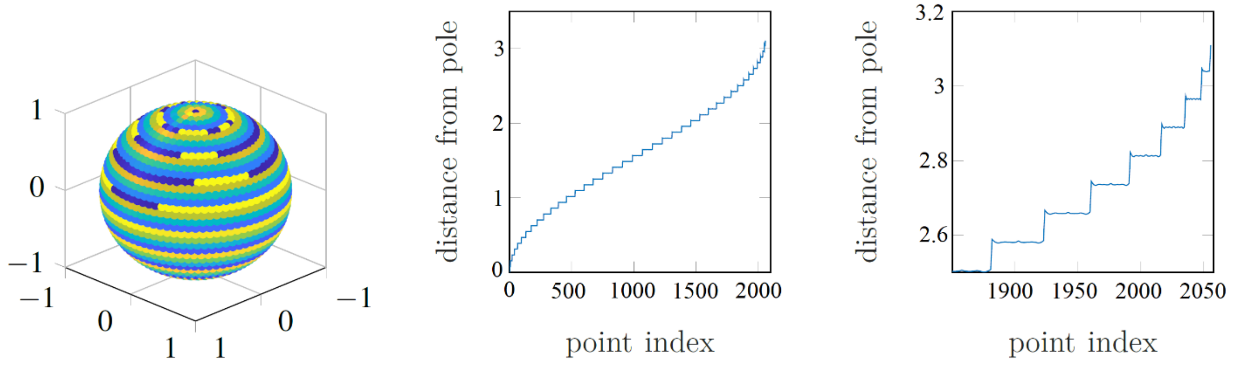
$$O_x = \frac{1}{2}\frac{(v_2)_x^2 + T_1^2 - T_2^2}{(v_2)_x}$$

$$O_y = \pm\sqrt{T_1^2 - \frac{((v_2)_x^2 + T_1^2 - T_2^2)^2}{4(v_2)_x^2}} = \pm\sqrt{T_1^2 - O_x^2}$$

$$T_3 = O_x \cdot e_x + O_y \cdot e_y - v_3$$

A challenge with this method is that there are always two possible virtual origins due to $\pm O_y$. In [4] it is stated that this is solved by calculating both distances and taking the larger value. However, there are situations where the smaller value is the correct one. This happens, presumably, mostly or only when P_3 is not in front of the wavefront but beside. Such a situation occurs when the distance of a point in the CVS is recalculated. To mitigate this issue in a simple way, the recalculated value for the distance T is only stored if it is smaller than the existing one.

2) Accuracy: The algorithm was tested on a sphere with equally spaced points as shown in Figure 1. The starting point, i.e., the point with distance T = 0 is chosen to be the north pole. The points are numerated such that one whole circle at constant θ is taken. Thus, plotting the distance over the index results in plateaus of constant distance as shown in Figure 1b and 1c.



(a) The colourmap was chosen to be repeating 10 times so that intra-ring fluctuations are visible.

(b) Staircase-distances emerging from the ring-after-ring-data-structure

(c) Zoom of the upper „stair“ shows oscillations (visible in intra-ring colour changes in (a))

Figure 1. Distances on a homogeneously sampled sphere. The index starts at the south pole and increases ring by ring until the north pole is reached. Distances are calculated from the south pole via the fast marching method.

III. Extracting the Geodetic Line

In the second section, the shortest distance from starting to ending point on the triangulated mesh is determined. In order to approximate the geodetic line, a line of shortest distance can be backtracked along the points in FVS. The real geodetic line, however, does not necessarily consist only of vertices but of points on edges of the triangles as well. In the following the first approximation of the geodetic line is denoted by Γ_0.

A. *Method of Minimum Distance*

The Method of Minimum Distance approximates Γ_0 with regard to the calculated distances *T*. It iterates the following procedure and can be modified through two different options:

1. The neighbor *N* of the previous point is determined which fulfills one of the following requirements:

1. Option 1: *N* has the lowest distance T_N of all provided neighbours.

2. Option 2: *N* is the point of neighbours for which the value of the distance T_N added to the distance from the previous point *p* is minimal

2. The resulting neighbor N is appended to Γ_0.

This method extracts the geodetic line very quickly but does not provide a good approximation, neither with Option 1 nor Option 2, especially when the grid is very uniform. Also, the points of the geodetic line are still only located on vertices. Therefore, the *gradient method* was implemented.

B. *The Gradient Method*

The gradient method provides an approach to extract the geodetic line dissociated from the vertices. To determine the direction in which the geodetic line propagates the gradient of the distance *T*, approximated with the three distances for each point in each triangle, is used.

1) *Approximation of the gradient in a triangle:*

The gradient in a triangle with vertices *i*, *j* and *k* is given by

$$(\vec{\nabla}T)_{(i,j,k)} = -\frac{\vec{n}}{|n|^2}\times\left(T_i\vec{e}_{jk} + T_j\vec{e}_{ki} + T_k\vec{e}_{ij}\right),$$

where

$$\vec{e}_{a,b} = \vec{x}_b - \vec{x}_a$$

are the vectors connecting the vertices *a* and *b* and $\vec{n}$ is the

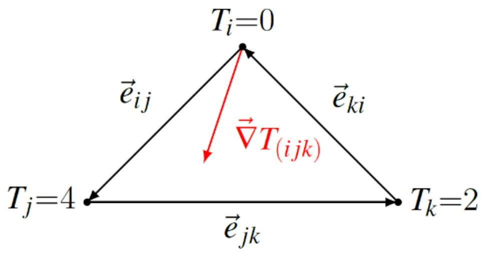


Figure 2. Three points (*i*, *j*,*k*) of a triangle with distance values *T* and the approximated gradient $(\vec{\nabla}T)_{(i,j,k)}$

surface normal of the triangle:

$$\vec{n} = \vec{e}_{ki} \times \vec{e}_{jk}$$

Note that the connecting vectors $\vec{e}_{ij}$, $\vec{e}_{jk}$ and $\vec{e}_{ki}$ are circular, i.e., that

$$\vec{e}_{ij} + \vec{e}_{jk} + \vec{e}_{ki} = 0$$

In Figure 2, a sketch of a triangle with its gradient is shown for an example set of distance values T_i, T_j, T_k.

2) *Extracting the Geodetic Line with the Gradient Method:*

The basic concept of the gradient method is to generate a line g for each triangle from the previous point p of the geodetic line and the gradient of T

$$g: \vec{x}(\lambda) = \vec{p} + \lambda \vec{\nabla} T_{(ijk)}$$

and to find its point of intersection with the edges of adjacent triangles. For the choice of edges to intersect g with, one has to consider whether the previous point p is on a vertex or an edge. If p is on a vertex, the following procedure is applied:

1. The negative gradients of the adjacent triangles are computed.
2. A triangle is determined whose negative gradient points into the triangle itself.
3. The line g is intersected with the edge of that triangle on the opposite side.
4. The point of intersection is added to Γ_0.

If no triangle is found whose negative gradient points into the triangle itself, the neighbour N with the smallest distance T to the previous point p is added to Γ_0.

If p lies on an edge, a different procedure is used:

1. The triangle which is adjacent to p and was not used for the prior calculation of p itself has to be identified.

2. The line g is intersected with the two remaining edges, if the negative gradient points into the triangle.

If the negative gradient does not point into the triangle, the previous p is moved to the vertex of the same edge that has the smaller distance T.

Special case: It might happen that p lies on a boundary edge. This case can be resolved by moving p to the vertex of the same triangle with a smaller distance T. If p lies on a boundary vertex, the above-mentioned procedure can be applied without further arrangements. As already mentioned, his is a special case. Therefore, this will not be considered in the further course.

3) *Performance of the Gradient Method*

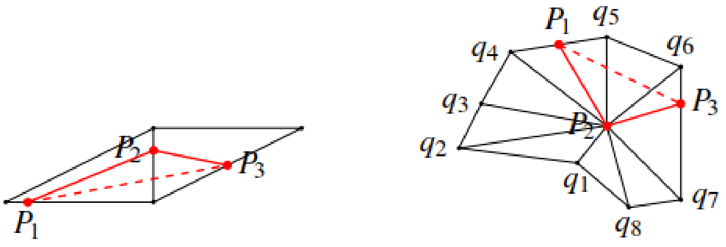
The algorithm approximates the real geodetic line in many test cases very precisely in accurate time. In case that real geodetic line runs near or along a line of edges without passing through several triangles or without changing the lane over the course of many points, the calculated geodetic line tends to stick to one lane and very late moves over to the other. This cannot be taken care of by the improvement algorithm which is described in the next section unless it is run for a lot more iterations than usual which is expensive. However, this special case is not problematic unless one wants to find the real geodetic line with even higher accuracy than already provided. For this, one could calculate the geodetic line and refine the triangulation around it to redo the whole calculation with the new triangulation until it converges.

IV. Improving the Approximation of the Geodetic Line

In the previous section we have generated an initial approximation Γ_0 for the geodetic line between two points on a triangulated mesh in three-dimensional space. As this is just a first approximation, an algorithm for improving Γ_0 is required. The improvement can be achieved by moving the points on vertices of the geodetic line along the edges of the mesh to shorten the length of Γ_0.

A. *Criterion for Improvement of the Geodetic Line*

According to [2] the shortest path is given by the straightest path for triangulated surfaces. ‘Straight’ is defined as follows: After taking all triangles that the approximation Γ_{i-1} passes through and unfolding them into a plane, the path Γ_i is the shortest when it is a straight line in the planar view. Therefore, the algorithm for improvement aims at straightening the path in the unfolded planar view.



(a) $P_{i,,j}$ lies on an edge.

(b) $P_{i,,j}$ lies on a vertex.

Figure 3. The two cases of a point of the geodetic Γ_i needing correction. For readability $P_{i,-1,,j}$, $P_{i,,j}$ and $P_{i,+1,j}$ have been replaced by P_1, P_2 and P_3, respectively. The dashed line denotes the corrected path.

B. The straightening algorithm

For this section the *i*-th version of the path is denoted as Γ_i and $P_{i,,j}$ the *j*-th point of the *i*-th path. For the following let $P_{i,,j}$ be the point to be corrected using the information about $P_{i,,j+1}$ and $P_{i,,j\,-1}$. The idea is to locally straighten the path by moving the central point of the three, i.e., $P_{i,,j}$. To ensure that the geodetic line converges and actually becomes shorter with each iteration, the updated $P_{i,+1,,j}$ for the updated path Γ_{i+1} is calculated using the points which have already been updated during this iteration, i.e., $P_{i,+1,j-1}$ instead of $P_{i,,j-1}$. For readability, we omit the "+1" in $P_{i,+1,j-1}$, but take care of it by only keeping one Γ stored and updating it with each step during each iteration.

There are always two cases to be considered: $P_{i,,j}$ lies on an edge or on a vertex as can be seen in Figure 3.

1) $P_{i,,j}$ lies on an edge:

If $P_{i,,j}$ lies on an edge, the following steps are needed to improve the line:

1. The two triangles adjacent to $P_{i,,j}$ are unfolded.

2. The point of intersection of the connecting line between $P_{i,,j\,+1}$ and $P_{i,,j\,-1}$ and the edge that $P_{i,,j}$ lies on are calculated. If the point of intersection does not lie between the two vertices of the edge, the closer vertex is chosen to be the corrected point instead in this case.

As an example, let us assume the three points $P_{1,}$ P_2 and $P_{3,}$ of which P_2 lies on an edge and is the point which has to be corrected. Since P_2 is on an edge, there are only two triangles adjacent to P_2. For this example, let us call the points of the first triangle *A*, *B* and *C*, and of the second triangle *B*, *C* and *D*. P_2 lies subsequently on the edge connecting *B* and *C*, P_1 lies in the triangle limited by *ABC* and P_3 lies in the triangle *BCD.* Applying the procedure explained above, first, the points *D* and P_3 have to be rotated around $\overline{CD}$ to be in the same plane as *ABC*. The rotated points will be denoted *D'* and P_3'. When defining *C* as the origin, every investigated point is moved by $-\vec{c}$, which is the position vector of *C* in the original coordinate system. The position vectors of the points P_3' and *D'* are then given by:

$$\vec{p}_3{}' \times= \vec{v}(\vec{v}\cdot\vec{p}_3)\ +\cos(\phi)\,(\vec{v}\times\vec{p}_3)\times\vec{v}+\sin(\phi)(\vec{v}\times\vec{p}_3)$$
$$\vec{d}' \times= \vec{v}(\vec{v}\cdot\vec{d})\ +\cos(\phi)\left(\vec{v}\times\vec{d}\right)\times\vec{v}+\sin(\phi)\left(\vec{v}\times\vec{d}\right)$$

Where the edge connecting the vertices *B* and *C* is given by

$$\vec{v}=\vec{b}-\vec{c}$$

And the angle by which the points have to be rotated by:

$$\phi=\pm\arccos\frac{(\vec{n}_1\times\vec{n}_2)}{|\vec{n}_1|\times|\vec{n}_2|}$$

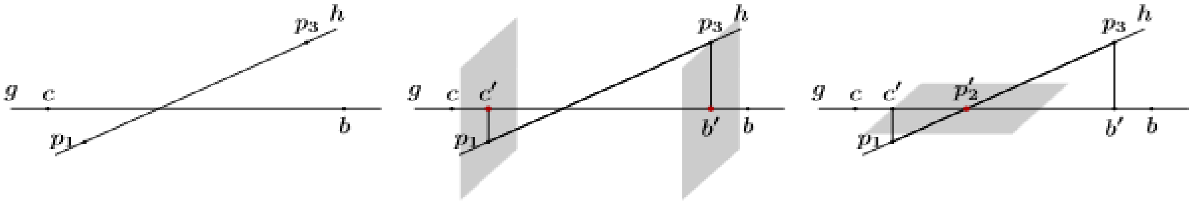
As the second step, the point of intersection of the edge $\overline{BC}$, in the following denoted as *h*, and $\overline{P_1, P_3{}'}$, in the following denoted as *g*, has to be calculated. The points *C'* and *B'* are found, which are defined as the points on *h* with the shortest distance to P_1 and P_3' which can be seen in Figure 4.

A plane at *C'* with normal vector $\vec{n}_2 = \vec{p}_1 - \vec{c}'$ can then be inserted as can be seen in Figure 4c. This plane can then be intersected with the line to find $\vec{p}_2$:

$$g{:}\,\vec{x}=\vec{p}_3+\ \lambda(\vec{p}_1-\vec{p}_3)=\vec{p}_2$$

With

$$\lambda=\frac{(\vec{c}'-\vec{p}_3)\cdot\vec{n}_2}{\vec{n}_2\cdot\vec{v}_2}$$

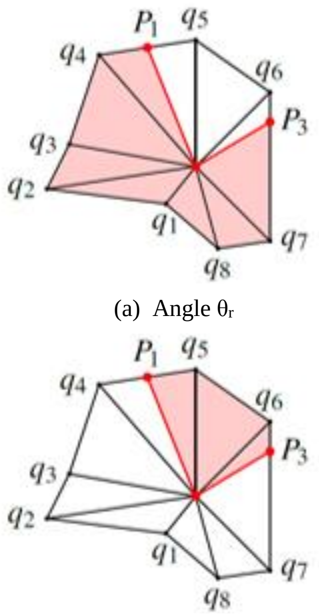


(a) Initial situation (two lines defined by four poitns)

(b) Two planes (perpendicular to *g*) added for finding *c'* and *b'*

(c) Plane (perpendicular to $\vec{p}_1 - \vec{c}'$) added for finding $\vec{p}_2$

Figure 4. Intersecting two lines



(a) Angle θ_r

(b) Angle θ_l

Figure 5. Notation of angles.

2) $P_{i,j}$ lies on a vertex

If the point that is to be corrected coincides with a vertex, the procedure becomes more complicated. Let S_k be the set of triangles that have $P_{i,j}$ as the central vertex, then several cases can be distinguished. Firstly, there are two simple cases which can be easily taken care of numerically:

a) If all three points ($P_{i,-1,j}$, $P_{i,j}$ and $P_{i,+1,j}$) are part of the same triangle, $P_{i,j}$ is removed from Γ.

b) If $P_{i,\ j+1}$ *or* $P_{i,j-1}$ lies on an edge that is not part of the boundary of S_k, it is removed from Γ.

For all other cases ($P_{i,\ j+1}$ and $P_{i,j-1}$ belong to two different triangles) the vertices around $P_{i,j}$ are sorted and the left and right hand angles, θ_l and θ_r, are calculated in order to characterize the vertex as can be seen in Figure 5. These angles are given by the sum of the central angles of the triangles which are obtained by splitting the star-like structure of S_k along the path $P_{i,-1,j} \longrightarrow P_{i,j} \longrightarrow P_{i,+1,j}$.

Three main cases can be distinguished:

1. $\theta = 2\pi$: euclidean

2. $\theta = \theta_l + \theta_r > 2\pi$: hyperbolic

3. $\theta < 2\pi$: spherical

These three cases are taken care of differently where θ is defined as left or right hand angle.

a) Euclidean: S_k can be unfolded isometrically. After unfolding, $P_{i,\ j+1}$ or $P_{i,j-1}$ are joined in the unfolded S_k and the intersections with the edges added to Γ.

b) Hyperbolic:

a. If θ_l and θ_r are greater than π: no correction is needed.

b. If θ_l and θ_r are smaller than π, that side of S_k is unfolded and $P_{i,\ j+1}$ as well as $P_{i,j-1}$ are joined in the same manner as in the Euclidean case.

c) Spherical: The part of S_k with smaller $\theta_{l/r}$ is unfolded and $P_{i,\ j+1}$ or $P_{i,j-1}$ are joined as in the Euclidean case.

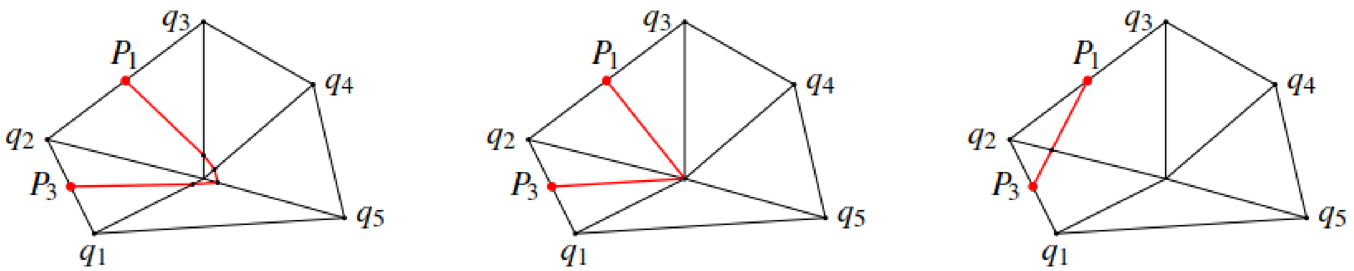
In all three cases the part of S_k with smaller $\theta_{l/r}$ has to be unfolded and the points of intersection have to be calculated.

In test runs, it was observed that points which are very close to vertices keep approaching the vertex which they are close to without coinciding and adopting its value. Therefore, every 10 iterations the path is scanned for points on Γ for which this might be the case. These points are moved to the vertex instead. All the following points that approach the same vertex are deleted from the path. This is necessary because otherwise curves in the path will never pass over a vertex. The effect of the scanning of the path and the movement of points to vertices is shown in Figure 6.

V. Exemplary Results

To test the capability of the gradient method and the straightening algorithm a geometry was chosen for which exact geodetic lines can be analytically computed for reference.

A plane with a half cylinder barrier is generated and the geodetic line between two points on either side of the half cylinder is calculated, first analytically, then using the



(a) The path is stuck at a vertex.

(b) After ten iterations all points close to a vertex are removed and replaced by one point which is directly located on the vertex.

(c) The next iteration corrects this point and the detour is removed.

Figure 6. Correction of points surrounding a vertex.

presented algorithm as shown in Figure 7.

The axis of the cylinder is set along the *y*-direction. Let us assume that the following parameters are given: the range of *x*, the step size *dx*, the range of *y*, the step size *dy*, the step size $d\phi$ for the angles on the cylinder, radius *r* of the cylinder and the displacement in *x*- direction of the cylinder axis *c* in the space. The cylinder is set in space such that there is translational invariance along the *y*-direction. Thus, the same *y*-vector can be used for each pair of *x* and *z*. The *x*-axis is separated into three parts. The first is an equally spaced vector that is limited by:

$$x = c - r - dx$$

The third is an equally spaced vector starting at

$$x = c + r + dx$$

The third part is the section in the middle of the mentioned ones and the one containing the half cylinder. This central part is given by:

$$x = c + r \cdot (-\cos(\phi))$$

For these three parts the *z*-coordinate is only unequal *zero* for the cylindrical part as can be seen in Figure 7. For the triangulation of the surface the Matlab function *delaunay* is used.

The surface can then be expressed in terms of a function *f* which is defined to be:

$$f: \mathbb{R}^2 \to \mathbb{R}$$
$$\text{i.e.}, f: (x, y) \to z(x, y)$$

This general generation of the test surface can be seen in Figure 7.

The analytical solution of the geodetic line running over the cylinder can be calculated as follows:

The starting and ending point, p_{start} and $p_{end,}$ of the desired geodetic line are set to be on different sides of the cylinder. The first step is then to find the point at which the geodetic line starts to climb the cylinder which is denoted as $\vec{\alpha}$ and the point at which the geodetic merges again to the plane which is denoted as $\vec{\beta}$ as can be seen in Figure 8a. To find these two points the half cylinder can be flattened which results in an isometric stretch of the grid in *x*-direction. The part which previously was on the right side of the half cylinder is then displaced by $r\pi - 2r$ as shown in Figure 8b, where shifted points are denoted with a prime (‘). Drawing a straight line between p_{start} and p_{end}’gives then the possibility to determine $\vec{\alpha}$ and $\vec{\beta}$ by the slope *m* of the drawn line. For this purpose, the slope is firstly calculated:

$$m = \frac{(\vec{p}_{end,y}{}' - \vec{p}_{start,y})}{(\vec{p}_{end,x}{}' - \vec{p}_{start,x})} = \frac{(\vec{p}_{end,y} - \vec{p}_{start,y})}{(\vec{p}_{end,x} + r(\pi - 2) - \vec{p}_{start,x})}$$

Let us call the *x*-component of the distance between p_{start} and the left bottom of the half cylinder Δα and the *x*-component of the distance between p_{end} and the right bottom of the half cylinder Δβ, which are then given by:

$$\Delta\alpha = c - r - p_{start,x}$$
$$\Delta\beta = -c - r + p_{end,x}$$

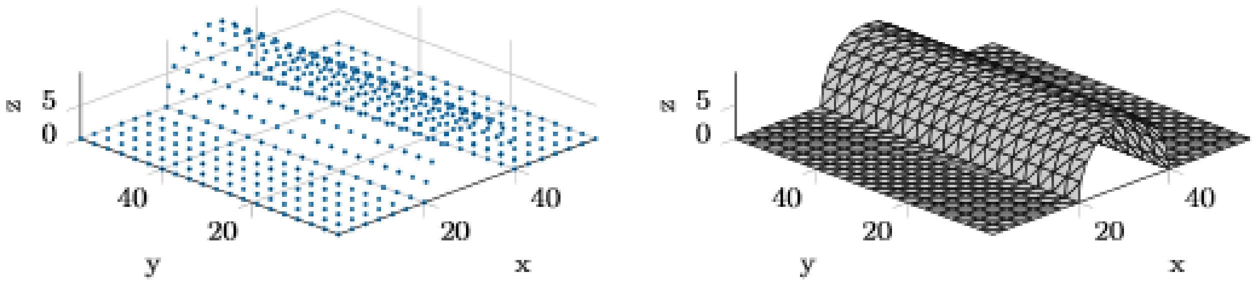
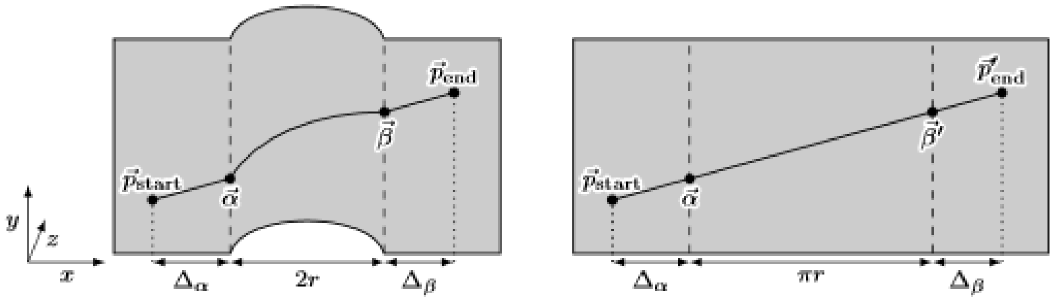


Figure 7. Scatterplot and delaunay triangulation of the test surface (Matlab).



(a) Sketch of a geodesic that goes over a cylindrical barrier. The points which it enters and leaves the barrier are highlighted ($\vec{\alpha}$ and $\vec{\beta}$)

(b) Stretching out the barrier to the right shifts $\vec{\beta}$ and p_{end} by $r\pi - 2r$

Figure 8. Flattening the half cylinder (Matlab Figure).

The coordinates of $\vec{\alpha}$ and $\vec{\beta}$ are subsequently given by:

$$\vec{\alpha} = \vec{p}_{start} + \begin{pmatrix} \Delta\alpha \\ m \cdot \Delta\alpha \\ 0 \end{pmatrix}$$

$$\vec{\beta} = \vec{p}_{end} - \begin{pmatrix} \Delta\beta \\ m \cdot \Delta\beta \\ 0 \end{pmatrix}$$

The distance in *x*,*y* and *z*-direction of the sections on the left-hand and right-hand side on the half cylinder can be easily calculated because the geodetic line is given by a straight line in a plane. The geodetic line in the central section can be analytically calculated as well:

$$y(\phi) = \alpha_y + \phi \cdot \frac{\beta_y - \alpha_y}{\pi}$$

In Figure 9, an exemplary result of the geodetic line analytically calculated is shown as a black line. The numerical result is indicated by red dots. Several aspects can be seen: Beginning at the ending point (on the right-hand side of the half cylinder) the distances of the other points are calculated by the FMM. These increase up to the starting point (on the left-hand side of the half cylinder). For clarity, the colour palette was chosen such that it is repeated five times.

In the left image the calculation is stopped after the extraction using the gradient method. For the right image the extracted geodetic line was improved by using the technique described in Subsection IV-B.

As can be seen, the straightening algorithm removed a few deviations visible close to the upper right end of the geodetic line. The straightening algorithm ran 50 times but most of the improvement was already achieved after five iterations.

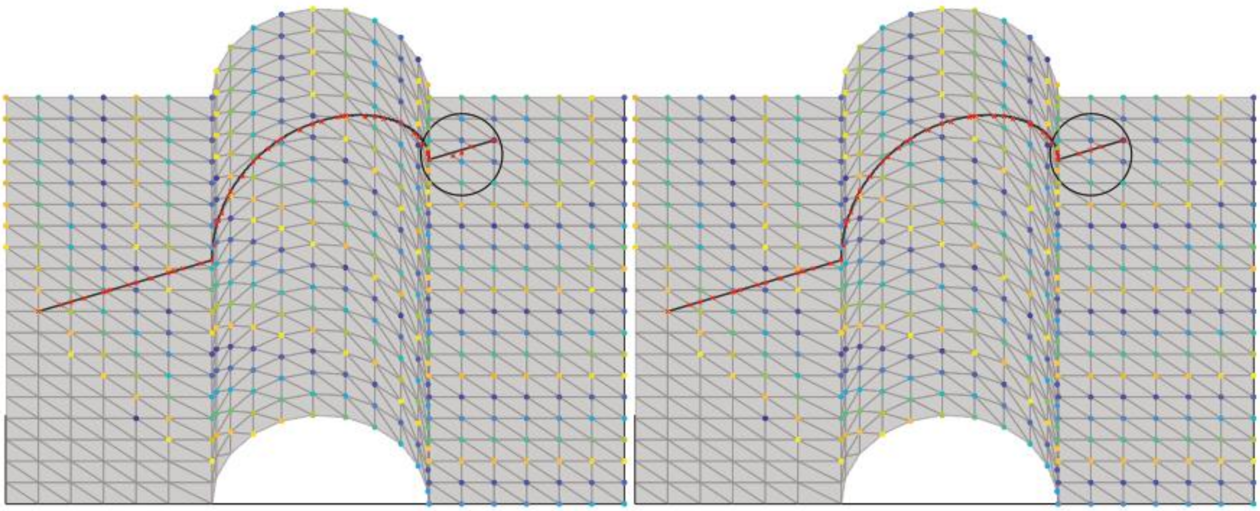


Figure 9. Result of fast marching method (both), geodetic extraction (left) and improvement algorithm (right)

VI. Creating Multiple Layers

When developing material structures, a given layer, which determines the structure of the desired object, has to be duplicated to ensure stability of the resulting object. For this, multiple considerations have to be taken into account which will be discussed in the following sections:

A) The layers, which extend the first one by being attached on it, do not have exactly the same size as the given one due to the thickness of the individual layers, which cannot be assumed to be infinitesimally small in practice. Thus, there are two general movements, which have to be considered. The first movement is the one, which the single investigated vertex is subjected to, i.e., the relative change of coordinates. The second one is the direction in which the surface moves as a whole. Here, the layers should shift to the *outside* of the surface, which has to be defined first.

B) In order to determine important specifications for the cutting pattern such as the thickness of the material layers and the width of the applied layers/material stripes, the point on the surface has to be found, which is subjected to the highest stress and thus experiences the highest strain in the whole layer. This point is, when regarding a surface without external loads being applied on it, the point of maximum curvature. At this point, the internal stress of the material is at its maximum. Material properties as well as manufacturing processes have to be adjusted in a way that the material at this point is still able to withstand the expected strains it might experience in its life cycle.

VII. Duplicating a Triangulated Layer

This section deals with the direction of the vertex shift, the direction of the surface shift and the resulting coordinates of additionally applied layers.

A. Direction of vertex shift

The direction $\vec{d}_i$, in which a single vertex moves, can be determined by the normal vectors $\vec{n}_i$ of the surrounding triangles T_i. Given are the edges of the triangles $\vec{e}_i$ and the corresponding vertices v_i. The normal vector $\vec{n}_j$ of a triangle is obtained by the cross product of two edges $\vec{e}_i$ depicted in vector form of the triangle T_j:

$$\vec{n}_j = \vec{e}_1 \times \vec{e}_2$$

If a point p_j is adjacent to i triangles T_i with normal vectors $\vec{n}_i$, then the direction $\vec{d}_i$ of the shift in p_j is given by the average of the normal vectors of the adjacent triangles as shown in 2-D in Figure 10.

$$\vec{d}_i = \frac{\sum(\vec{n}_i)}{i}$$

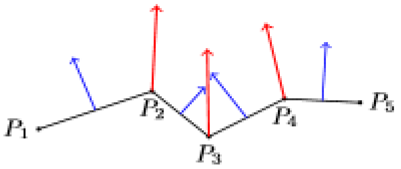


Figure 10. Calculation of normal vectors in points (red) from normal vectors in triangles (blue).

B. Direction of surface shift

1) The 'outside' of the surface

The outer face of protection gear, which is particularly aimed at in this project, is predominantly defined by convex structures such as shield, helmets and vests. The outside is subsequently generally defined as the space above the convex curvature of the surface. The problem, which arises, is that the orientation of the surface, or more specifically, of the order and direction of edges is not given by the *delaunay* triangulation in Matlab. Therefore, the calculation of the normal vectors is not necessarily right-handed which results in normal vectors that can point inside the surface when calculated with a left-handed system and outside the surface when calculated with a right-handed system. Hence, the normal vectors have to be oriented in a way that they all point in one direction which can then be easily adjusted to the outside of the surface.

Owing to the assumption that there are no large kinks in the triangulated surface, one can say that the projection of the normal vector of any triangle on the normal vectors of each of the adjacent triangles, i.e., the scalar product c_i, has to be positive when the normal vectors of the triangles point in the same direction. For this concept to work, the triangulation has to be fine enough, so that on a *convex* surface normal vectors of adjacent triangles, which point on the same side of the surface, have a positive scalar product without exception.

$$c_i = \vec{n}_i \cdot \vec{n}_{i+1} >! 0$$

2) Direction of the surface

To shift all vertices in the same direction, all normal vectors have to point in the same direction of the surface. At this point of the procedure, it does not matter, whether they point inside or outside the surface. The specified assumptions and calculations result in the following procedure to set the direction:

1) One arbitrary reference triangle T_r is determined, whose direction of the relative normal vector gives the reference normal vector for all other triangles (points to inside or outside)

2) It is iterated through all adjacent triangles of T_r and the projections of the normal vectors are calculated, i.e., the scalar product, of the normal vectors of the adjacent triangles on the reference normal vector.
3) If the scalar product is smaller than zero, the relative normal vector of the adjacent triangle has to be flipped, i.e., multiplied with "–1".
4) It is iterated for all triangles (without step a)) such that there is always one triangle in the neighborhood, which has already been checked on its direction. This triangle then sets the new reference normal vector.

At the end of this procedure, all normal vectors point on the same side of the surface, either the inside or the outside of the surface.

For the following principle, it is additionally assumed that the coordinate system is set such that the convex part of the structure points in positive *z*-direction.

To check, whether the normal vectors point inside or outside the structure, it is iterated through all normal vectors and searched for the one with the largest z-component, which is denoted as $\vec{z}_{max}$. The vector $\vec{z}_{max}$ should be the one of the triangle on top of the convex curvature of the surface, e.g., on top of the helmet. This works even with a rotated surface as long as the convex curvature points upwards. Owing to this, the scalar product of the unit vector in *z*-direction $\vec{e}_z$ and $\vec{z}_{max}$ can be used to determine the orientation on now all normal vectors because they all point on the same side of the helmet.

1) If the scalar product is positive, the normal vectors of all triangles point outside the surface. No arrangement has to be made in this case.

$$c_i > 0 \;\rightarrow\; \vec{n}_{i\,,final} = \vec{n}_i$$

2) If the scalar product is negative, the direction of all normal vectors has to be reversed, i.e., multiplied with "–1" so that they point outwards.

$$c_i < 0 \;\rightarrow\; \vec{n}_{i\,,final} = \vec{n}_i \;\cdot (\text{-}1)$$

The result at this point of the algorithm for multiple layers are the directions in which each point has to be shifted. Due to numerical round-off errors, the normal vectors might not all be of length 1, which is addressed in the next section.

3) Shifting the vertices

The vertices v_i of the triangulation can now be shifted *n*-times by the distance *s* in direction $\vec{d}_i$, where *n* is the number of required additional layers and *s* is the thickness of the applied layer. The coordinates for the new layers are denoted as X_{n+1} which is given in matrix form. At this stage, distance *s* is manually given. As mentioned above, the normal vectors may not be of length "1", which is why the distance *s* is divided by the length of the normal vector l in the following equation. X_n and the summand are of the same dimension as X_n stores all coordinates of *m* points in *m* rows and three columns for the corresponding *x*-, *y*- and *z*-components and the summand denotes the shift in *x*-, *y*- and *z*- direction stored in three columns for *m* points stored in *m* rows.

$$X_{n+1} = X_n + [\; n \cdot \vec{d}_i \cdot \frac{s}{l_i} \;]$$

By this, the given layer can be shifted arbitrarily often while maintaining its original shape as can be seen in Figure 11.

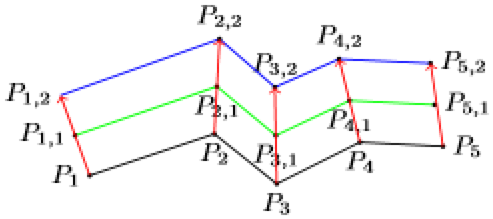


Figure 11. First layer (black) shifted two times (green and blue lines) in the direction of the normal vectors in the points (red).

VIII. Point of Maximum Curvature

The curvature of points can be calculated with an approach given in [2]. In this approach, every single triangle is investigated independently of the surrounding triangles. The curvature directly in the vertices is then set to zero. However, in this project, it is assumed that the number of elements over the whole surface, i.e., the number of triangles, is so large that the triangles can be approximated with planes such that there is no curvature within one individual triangle but the convex shape of the whole structure is obtained by putting the triangles together with different angles of inclination. The triangulated mesh is assumed to be so fine that the differences in angles of inclination of two adjacent triangles never exceeds 90 degrees. Consequently, scalar products of the normal vectors of adjacent triangles are always positive. Therefore, these scalar products of the normal vectors of adjacent triangles can be used to approximate the curvature by the difference in inclination of two adjacent triangles. The 'curvature' of the surface in a given vertex can then be determined by averaging over the scalar products of the adjacent triangles. The smallest scalar product then results in the point of maximum curvature, which can be used in future research.

IX. Conclusion and Future Work

An algorithm for calculating a geodetic line on a given surface, a technique for its further improvement and the numerical derivation of multiple layers on a surface are described. The goal was to derive an accurate numerically determined geodetic line as well as duplication of a given layer. Further steps could feature an extension of the algorithm for geodetic lines, such that several geodetic lines on one surface can be found by iterating over the algorithm. To further improve, analyse and straighten the geodetic line, the unfolding of surfaces with the least distortion could be investigated and automatized [1]. For this, the point of maximum curvature, which was derived in the last section, can be used as the point of investigation because it faces the largest distortion when being flattened. As a reference for the distortion the calculation of metric and angular change of the flattened and convex surface [5] or the elastic potential [6] could be used. At this stage the aim is that the geodetic lines, i.e., the cutting pattern, should be set in a way that the distortion when flattening is minimised because the material is cut in the 2-D plane and is then formed into the desired state or attached on a convex 3-D object. For this process of flattening, the approach given in [7] can be referred to which also presents an approach on how to measure the amount of distortion.

Furthermore, thresholds values for the thickness of layers and width of material stripes could be set with reference to the point of maximum curvature depending on material properties such as Young's Modulus and Poisson's ratio. Both values should never exceed a threshold where too much stress is applied on the material when bending over the maximum point of curvature. This threshold value for the maximum applicable stress could be set to be the ultimate tensile stress when working with prestressing. In [8] basis properties of elasto-mechanical properties of materials with and without prestressing as well as distortion are discussed which could be used as a starting point for future research in this topic.

Additionally, the change in accuracy dependent of the number of elements in the triangulated mesh could be investigated in order to define and optimize the relation between these two quantities.

Summarized, future work could contain research about appropriate thresholds, materials and manufacturing specifications with the aim to reinforce a given structure.

References

[1] A. Pestalozza, S. Weichert, A. Ramezani, H. Rothe, Generation of a Geodetic Line on any Given Surface, The 10th international Conference on Advances in System Simulation: Simulation Based Development of Defense Systems, 2018

[2] J. M. F. Linhard, Numeric-Mechanical Investigation of the Planning Process of Membrane Structures, PhD thesis, Technische Universität München, 2009.

[3] D. Martínez, L. Velho, and P. C. Carvalho, Computing geodesics on triangular meshes, Computers & Graphics 29(5):667 – 675, 2005.

[4] M. Novotni und R. Klein, Computing geodesic distances on triangular meshes, Journal of WDCG, 10(1-2):341-347, 2002.

[5] L.Ju, J. Stern, K. Rehm, K. Schaper, M. Hurdal and D. Rottenberg, Cortical Surface Flattening Using Least Square Conformal Mapping With Minimal Metric Distortion, 2nd IEEE International Symposiu on Biomedical Imaging: Nano to Macro (IEEE Cat No. 04EX821), 2004.

[6] K.-U. Bletzinger, A. Widhammer, Variation of Reference Strategy-A novel approach for generating optimized cutting patterns of membrane structures, Procedia Engineering, 2016.

[7] B. Lévy, S. Petitjean, N. Ray and J. Maillot, Least Squares Conformal Maps for Automatic Texture Atlas Generation, ISA (Inria Lorraine and CNRS), 2002.

[8] D. Ströbel, The Usage of the Adjustment Calculation of Elasto-Mechanical Systems, 1997.