Generation of a Geodetic Line on Any Given Surface

Anna von Pestalozza, Stefan Weichert, Arash Ramezani, Hendrik Rothe
Chair of Short-Time Dynamics
University of the Federal Armed Forces
Hamburg, Germany
Email: pestalozza@hsu-hh.de, s-weichert@web.de, ramezani@hsu-hh.de, rothe@hsu-hh.de

***Abstract*—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.**

Keywords-Geodetic Line; Surface Analysis; Materials Technology.

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. Furthermore, the starting point of the presented research is the approach given in [1] for finding geodetic l ines between two points. The main idea is to successively calculate distances from the starting point which is improved by the fast marching method. In the following, an algorithm for extracting the geodetic line as well as for further improving it is derived.

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.

The paper is divided in five 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. The paper ends with a concluding paragraph in Section V and an outlook to future work on this project.

II. CALCULATION OF SHORTEST DISTANCES ON A TRIANGULATED MESH

In this paper, the procedure of the Fast Marching Method (FMM) is used [3] 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) Fixed vertex set (FVS): 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

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-672-9

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 ν_i with $\nu_i = P_i - P_1$, in particular $(\nu_3)_x$ and $(\nu_3)_y$, i.e., the projections of ν_3 onto the new basis vectors, which are calculated as follows: To change the default, adjust the template as follows:

a) One point is set as the origin (P_1):

$$\nu_i = P_i - P_1$$

b) The coordinate system is transformed where:

$$e_x = \frac{(p_2 - p_1)}{|p_2 - p_1|} = \frac{(\nu_2)}{|\nu_2|}$$

$$e_y = \frac{(\nu_3 - e_x \cdot (e_x \cdot \nu_3))}{|(\nu_3 - e_x \cdot (e_x \cdot \nu_{3)})|} = \frac{(\nu_3 \cdot |\nu_2|^2 - \nu_2 \cdot (\nu_2 \cdot \nu_3))}{|(\nu_3 \cdot |\nu_2|^2 - \nu_2 \cdot (\nu_2 \cdot \nu_3)|}$$

c) The distance of ν_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:

$$O_x = \frac{1}{2}\frac{(\nu_2)_x^2 + T_1^2 - T_2^2}{(\nu_2)_x}$$

$$O_y = \pm\sqrt{T_1^2 - \frac{((\nu_2)_x^2 + T_1^2 - T_2^2)^2}{4(\nu_2)_x^2}} = \pm\sqrt{T_1^2 - O_x^2}$$

$$T_3 = O_x \cdot e_x + O_y \cdot e_y - \nu_3$$

A challenge with this method is that there are always two possible virtual origins due to $\pm O_y$. In [3] 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 (see Figure 1 b and c).

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:

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

b) 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

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

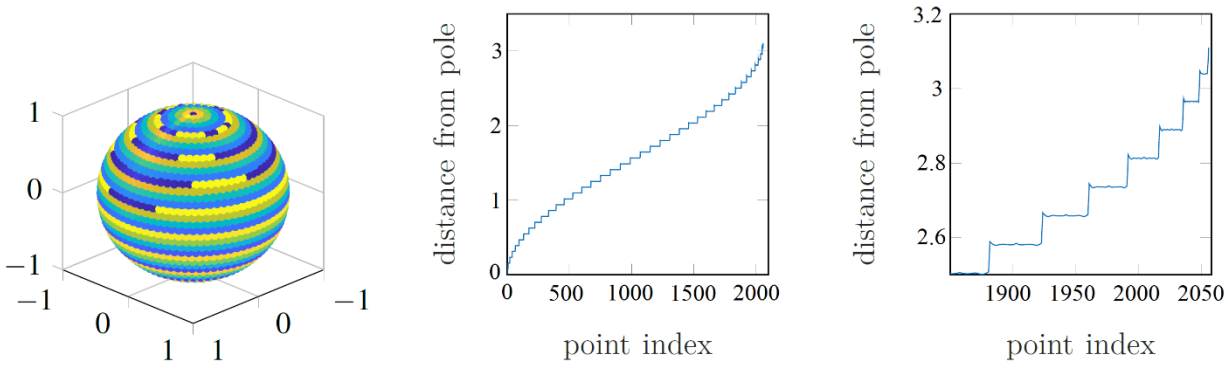
where

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

are the vectors connecting the vertices *a* and *b* and $\vec{n}$ is the surface normal of the triangle:

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

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-672-9



(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 "stairs" shows oscillations (visible in intra-ring colour changes in (A).)

Figure 1. Distances on 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.

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.

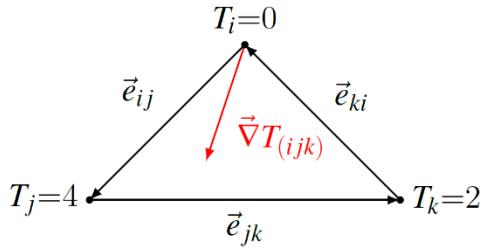


Figure 2. Three points (i, j, k) of a triangle with distance values T and the approximated gradient $(\vec{\nabla}T)_{(i,j,k)}$

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 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.

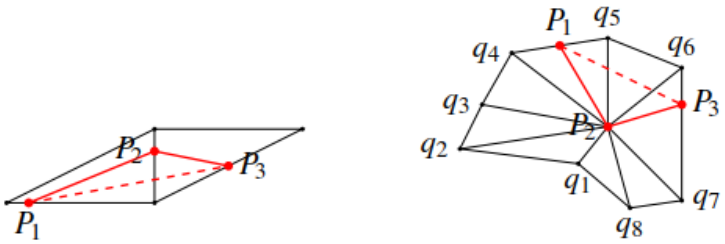
If the negative gradient does not point into the triangle, the previous p is moved to the vertex of the same edge that has the smaller distance T.

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, this 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

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-672-9



(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 geodesic 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.

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.

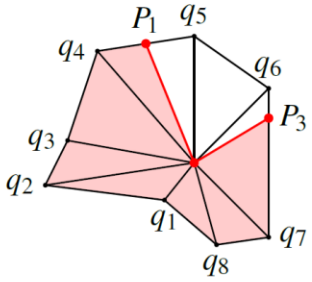
B. The straightening algorithm

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.

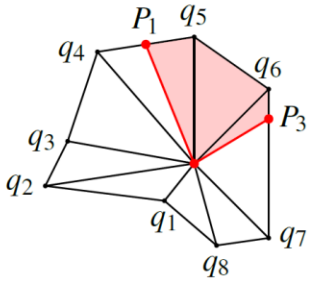
There are always two cases to be considered: $P_{i,j}$ lies on an edge or on a vertex as can be seen in Figure 3.

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

If $P_{i,j}$ lies on an edge, the two triangles adjacent to $P_{i,j}$ are unfolded. 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.



(a) Angle θ_r



(b) Angle θ_l

Figure 4. Notation of angles in the star-like structure of S_k

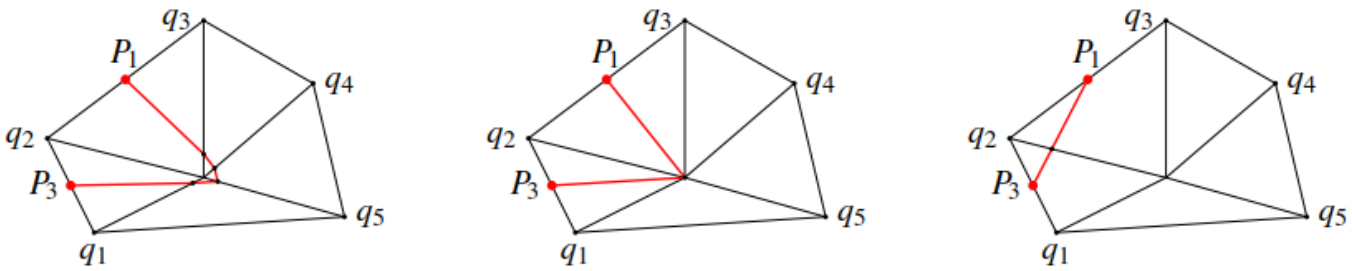
2) $P_{i,j}$ lies on a vertex

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.

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-672-9



(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 5. Correction of points surrounding a vertex

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 4. 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 $P_{i,-1,j} \rightarrow P_{i,,j} \rightarrow P_{i,+1,j}$.

Three main cases can be distinguished:

a) $\theta = 2\pi$: euclidean

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

c) $\theta < 2\pi$: 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 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 5.

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 presented algorithm as shown in Figure 6.

In Figure 6, 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. The black line shows the analytically calculated geodetic line. The red crosses show the points of the geodetic line which was calculated with the above-described algorithm.

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 5 iterations.

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-672-9

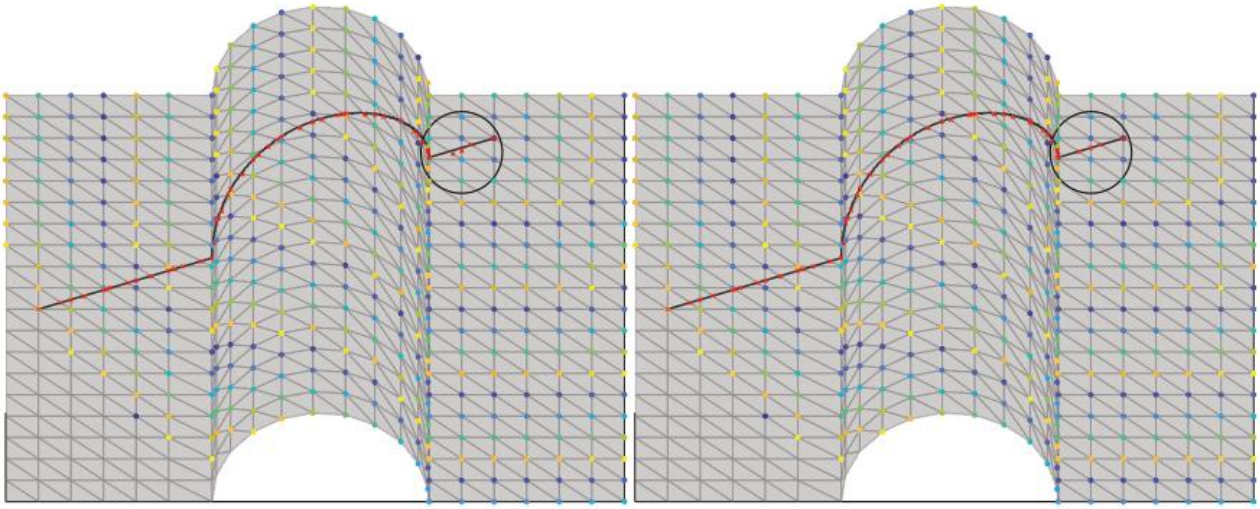


Figure 6. Result of fast marching method (both), geodesic extraction (left) and improvement algorithm (right).

VI. Conclusion and Future Work

An algorithm for calculating a geodetic line on a given surface and a technique for its further improvement are described. The goal was to derive an accurate numerically determined geodetic line. Further steps could feature an extension of the algorithm, 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 automatised. Moreover, 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.

References

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

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

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

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-672-9