02
Review paper I Прегледни научни рад
DOI 10.7251/STP2215011L
ISSN 2566-4484

Daniel Lordick, TU Dresden, daniel.lordick@tu-dresden.de

CONSTRUCTION OF SKEW RULED SURFACES BY MANIPULATING CONTROL LINES

Abstract

When it comes to doubly curved surfaces in construction, especially shell structures, the skew ruled surfaces play a prominent role, because they are statically efficient and relatively easy to produce [1]. However, in order to fully exploit the design potential of this class of surfaces, conventional CAD programs do not provide adequate tools. From the priority program 1542 by the German Research Foundation (DFG) named *Concrete light*, an add-on is presented [2] that facilitates parametric design with ruled surfaces on the basis of line geometry and also enables a connection to finite element methods.

Keywords: Line Geometry, ruled surface, interpolation algorithms, parametric design

КОНСТРУКЦИЈА ЗАКРИВЉЕНИХ ПРАВОИЗВОДНИХ ПОВРШИ МАНИПУЛАЦИЈОМ КОНТРОЛНИХ ЛИНИЈА

Сажетак

Када је ријеч о двоструко закривљеним површима у грађевинарству, посебно о конструкцијама љуске, завојне правоизводне површи имају значајну улогу, јер су статички ефикасне и релативно лаке за израду [1]. Међутим, да би се у потпуности искористио потенцијал пројектовања ове групе површи, уобичајени програми CAD не пружају одговарајуће алате. Из приоритетног програма 1542 Њемачке истраживачке фондације (ДФГ) под називом *Concrete light*, представљен је додатак [2] који олакшава параметарско пројектовање са правоизводним површима на основу линијске геометрије и такође омогућава повезивање са методама коначних елемената.

Кључне ријечи: линијска геометрија, правоизводна површ, интерполациони алгоритми, параметарско пројектовање.

1. RULED SURFACES

1.1. RULED SURFACES AS A SUBSET OF NURBS SURFACES

In common CAD programs, ruled surfaces occur as a subgroup of *Non-Uniform Rational Basic Splines* (NURBS). A NURBS surface is a ruled surface if the parameter curves (*isocurves*) in at least one direction have the algebraic order 1 (degree 1, linear), i.e. if they are straight line segments and thus the *generators* (rulings) of the ruled surface. This can be achieved in two ways: Either one generates the NURBS surface by connecting exactly only two profile curves with a loft, or by placing the loft surface through a set of exclusively straight lines.

But, ruled surfaces are by no means unambiguously and traceably defined by this procedure. In the first case, the generators always connect corresponding parameter points on the profile curves. However, the distribution of the parameter points depends on the parameterization of the profile curves and is therefore largely arbitrary. This degree of freedom reflects the fact that ruled surfaces are only uniquely determined by three curves. Two curves allow for an infinite number of ruled surfaces and the CAD user initially has no control, which surface the NURBS algorithm offers. To obtain a desired ruled surface, additional processing is essential. For example, it may be required that all generators be parallel to a directional plane, which leads to a conoidal surface. It is then necessary for the user to draw a set of such generators and to loft the surface through these straight lines as described above.

If, on the other hand, a set of straight lines is given, the shape of the ruled surface depends strongly on the number of given lines. The NURBS algorithm interpolates the distances as if control points for the second set of parameter curves were given. The result is in general not predictable. In order to approximate the boundary curves of the surface to the first profile curves again, the generators must be placed sufficiently close to each other.

1.2. RULED SURFACES FROM A LINE GEOMETRY PERSPECTIVE

We are used to understand and handle our three-dimensional space by defining points with three coordinates in the Cartesian coordinate system. However, if we consider the straight lines as basic elements instead of the points, we find out that at least four coordinates are needed to define a straight line: if you first define a starting point in the x-y-plane, this binds two coordinates. A further coordinate gives - for example with an angle - the direction in the x-y-plane and a fourth one the gradient. The manifold of all straight lines in the three-dimensional space is therefore four-dimensional. This makes the handling of straight lines considerably challenging.

From a mathematical point of view, one approach is to embed the four coordinates again as points in a model space of higher dimension. In the present case, the so-called Study sphere is used for this purpose, which, among other things, is also suitable for describing the motions of six-axis robots [3]. Now, each point on the four-dimensional Study sphere corresponds to an oriented straight line in three-dimensional space. The Study sphere can be thought of as a unit sphere where each point on the surface signifies a straight line direction. Attached to the point of the sphere is another vector tangential to the sphere, which encodes the distance of the straight line to the origin of the coordinates. This point model can now be used for elegant calculations based on dual numbers. A detailed description can be found in [4].

The calculations on the Study sphere were adapted for civil engineering under the aspect of lightweight constructions in the research project "Force-adaptive discretization of lightweight concrete elements by means of line geometric modeling" at the TU Dresden and applied in the follow-up project together with Mike Schlaich at the TU Dresden under the title "Lightweight Concrete Structures Based on Line Geometry". In particular, the task was solved, how, after the interpolation algorithms on the Study sphere, the actually infinitely long straight lines can be reasonably restricted to the relevant area in three-dimensional space. One realization is:

2. THE ADD-ON LINEGEOMETRY

LineGeometry is an add-on for *Grasshopper*, which in turn serves as a plug-in for the CAD software *Rhinoceros 3D*. In recent years, Grasshopper has become an extremely popular tool for parametric modeling of three-dimensional objects. *LineGeometry* fits seamlessly into this working environment.

2.1. INSTALLATION

The add-on is available for free download with additional material here: [5]. In the package you will find the add-on file with the extension *.gha, which can be dragged and dropped onto the workspace of Grasshopper (canvas). Then the functions can be called via the LineGeometry tab. The website also provides sample files and additional information.

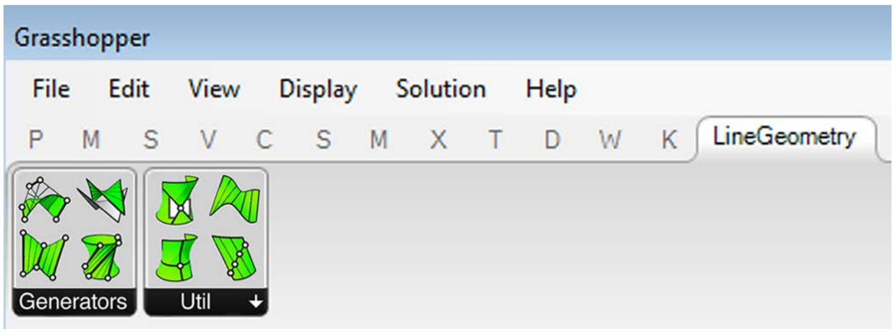

Figure 1. *The LineGeometry add-on in Grasshopper.*

The functions of *LineGeometry* are distributed over two blocks. The first block (*Generators*) contains the tools for creating ruled surfaces as a set of generators. The second block (*Util*) contains auxiliary functions.

2.2. GENERATION OF RULED SURFACES (GENERATORS)

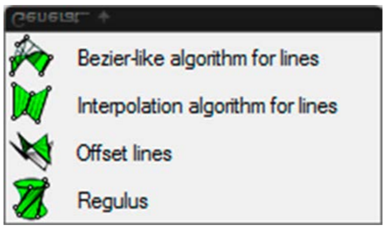
Bezier-like algorithm for lines
Interpolation algorithm for lines
Offset lines
Regulus

Figure 2. *Generators in the LineGeometry add-on.*

The starting point for the generation of ruled surfaces with *LineGeometry* is always a selection of oriented control lines. The orientation of the straight lines is given by the parameterization during drawing, where the end point drawn first is also the start of the parameterization. In case of problems, the orientation of individual lines can be corrected at any time with the tools of Rhinoceros or Grasshopper (e.g. with the Grasshopper component *Flip Curve*). For the following illustrations, four mutually skewed lines have been drawn in Rhinoceros and are available in Grasshopper via the *Curve component* (not *Line*!) under the name *Crv - four oriented lines*. The control lines are labeled with the numbers *0* to *3*. In the illustrations of the Grasshopper canvas, the components from the *LineGeometry* add-on are each highlighted by a green background.

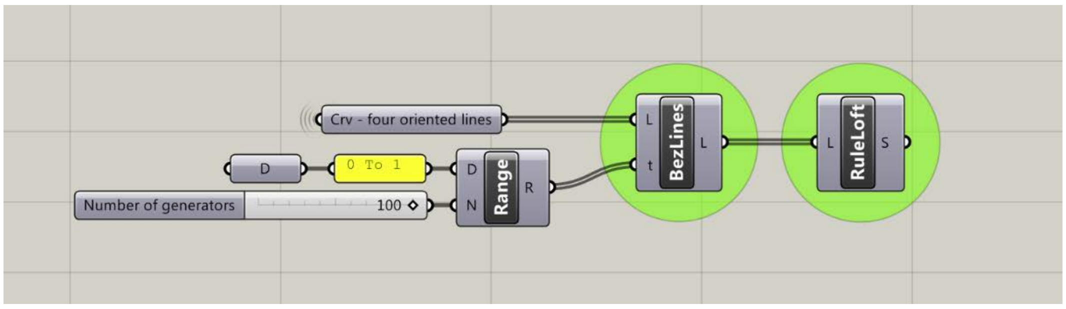

Figure 3. *BezLines in the LineGeometry add-on.*

The first component is called **Bezier-like algorithm for lines** (*BezLines*) (Fig. 3). It transfers the Bezier algorithm for splines to the Study sphere and calculates a set of generators from the given control lines. The generators belong to a ruled surface interpolating the control lines. In general, only the first and the last control line are contained in the ruled surface. The remaining straight lines are *control lines* in the same way that for splines the vertices of the control polygon control the spline. The input *L* (*Control lines*) accepts the control lines as a list and the input *t* (*Range of* *parameters*) accepts any number of values between 0 and 1 (*Domain D: 0 To 1*, also in the following, if not stated otherwise). Each value t corresponds to one generator. The values *0* and *1* coincide with the first and the last control line. At the output *L* (*Resulting generators of the ruled surface*) are the generators of the ruled surface (Fig. 4).

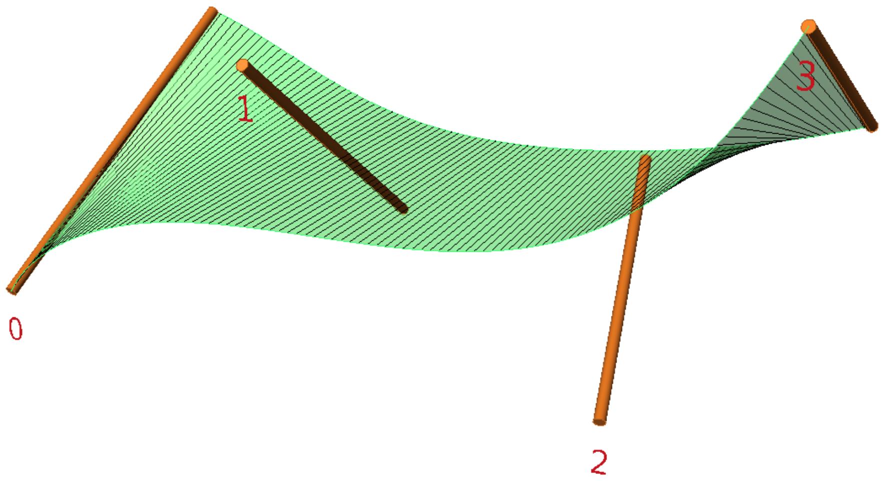

Figure 4. *Ruled surface from four control lines with BezLines.*

If the value domain *0* to *1* is exceeded, the surface is extrapolated with more or less reasonable results. If only two control lines are passed at input L, the result of the interpolation is a helicoid (Fig. 5). The helicoid is the only ruled surface which at the same time is a minimal surface (apart from the trivial case of the plane). In this case, the contrast to the loft functions of Grasshopper and Rhinoceros becomes obvious, because there a loft between two skew lines always results in a hyperbolic paraboloid.

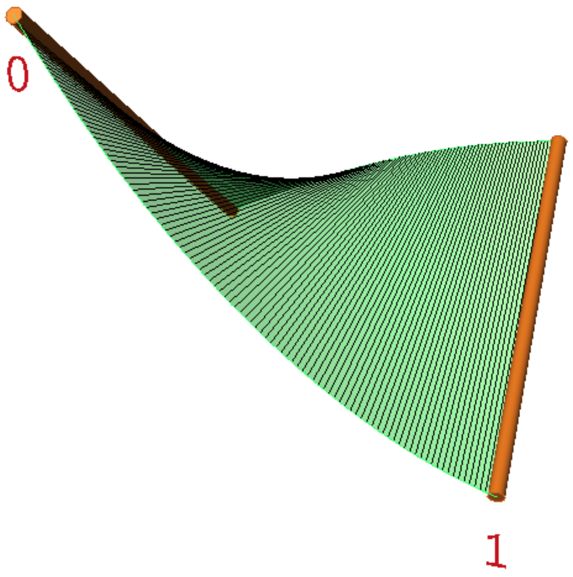

Figure 5. *Helicoid as interpolation of two control lines, that is, two points on the Study sphere.*

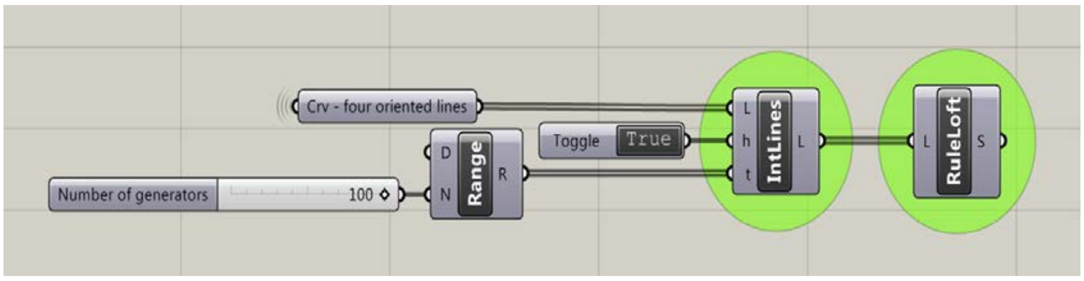

Figure 6. *IntLines in the LineGeometry add-on.*

The second component is called **Interpolation algorithm for lines** (*IntLines*) (Fig. 6). For this component, the Aitkin algorithm for drawing splines was transferred onto the Study sphere. The difference to BezLines is that now the control lines are actually contained in the interpolating ruled surface (Fig. 7).

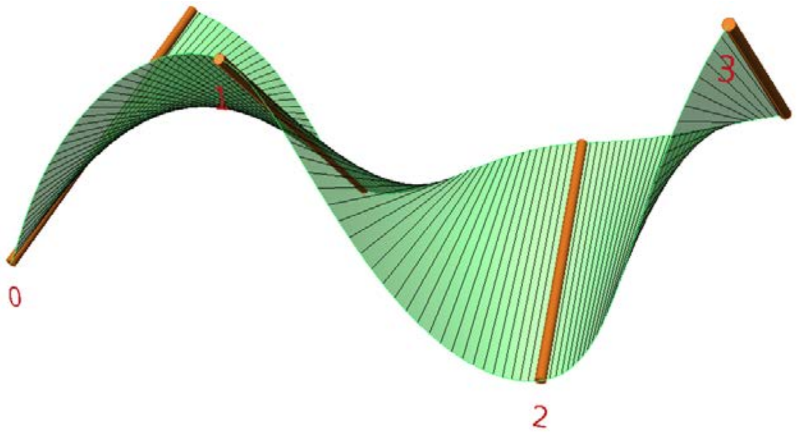

Figure 7. *Ruled surface from four control lines with IntLines.*

The additional input *h* (*Enable piecewise linear interpolation*) offers the possibility to connect the control lines piecewise with helicoids (Fig. 8). *Linear interpolation* thus means that geodesic ("shortest") paths are chosen between those points on the Study sphere, which correspond to the control lines. The *BezLines* component could be used to accomplish the same thing by feeding the control lines in pairs. Here, piecewise interpolation is easily achieved if *h* is *1* or *True*. With *0* or *False IntLines* provides the smooth interpolation with the algorithm derived from Aitken.

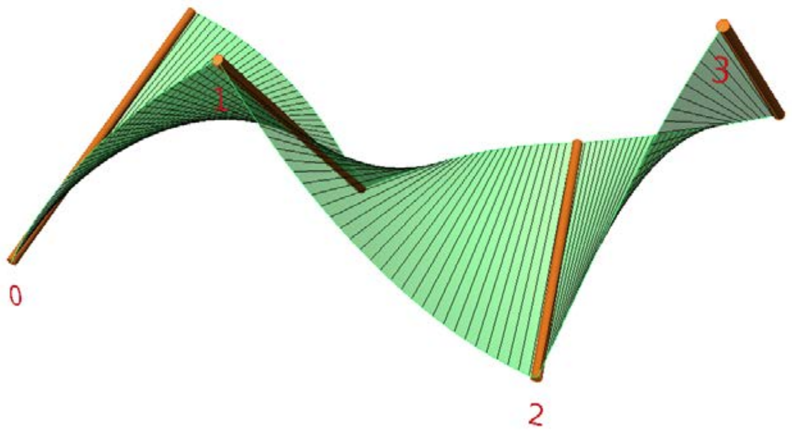

Figure 8. *Piecewise linear interpolation with IntLines.*

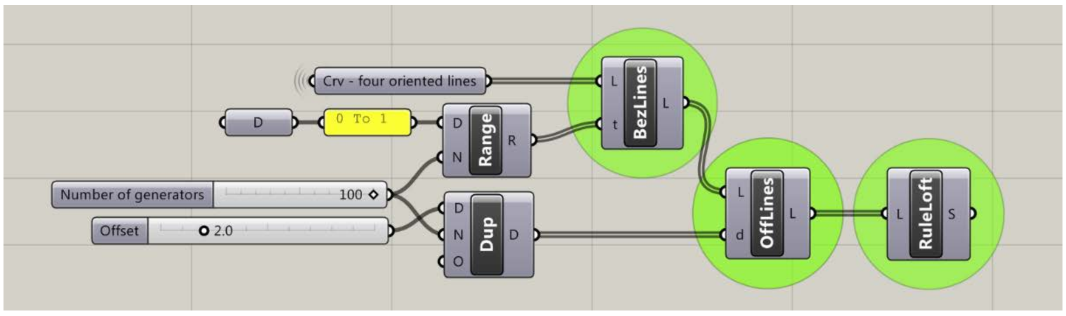

Figure 9. *OffLines in the LineGeometry add-on.*

The third component is called **Offset lines** (*OffLines*) (Fig. 9). As is well known, the parallel surface of a skew ruled surface is not a ruled surface any more. However, especially in civil engineering it is often required, to define both sides of a structural element as ruled surfaces. *OffLines* provides an alternative to the standard surface offset by generating a "parallel" ruled surface from a ruled surface (Fig. 10). The component requires as input in *L* (*Lines*) a discrete set of generators of a ruled surface and for each generator the desired distance in *d* (*Distance value for each line*). It is therefore necessary, if the distance is to be the same everywhere, to duplicate the desired value with the Grasshopper component *Duplicate Data* (*Dup*) a corresponding number of times. Again, the ruled surface created with *OffLines* does not have the same distance from the original surface everywhere. Depending on the curvature of the surface, the distance of the generators in the center of the surface is significantly smaller than at the edge. For the calculation of the offset lines, only the end points of the generators are offset normal to the initial surface and then connected again with a straight line. By the way, as a special feature, the component allows a different distance value to be taken into account for each generator. Thus, functions for manipulating the distances can be used here, for example depending on possibly existing load cases.

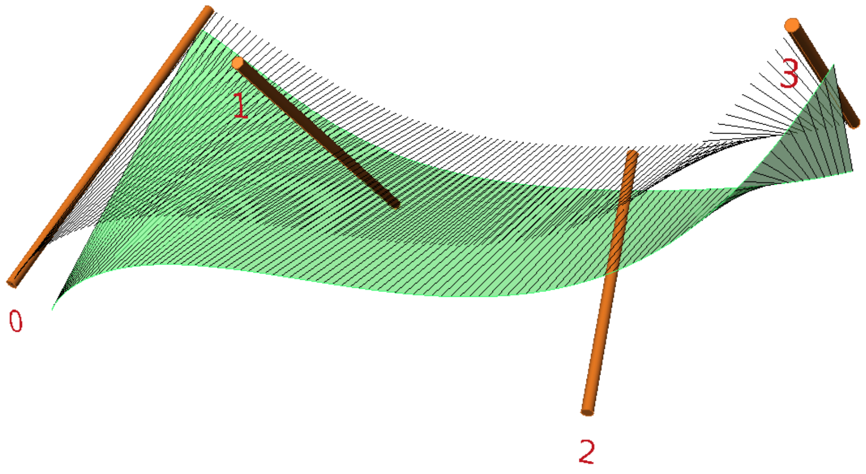

Figure 10. *"Parallel" ruled surface to Fig. 4 using OffLines (in green).*

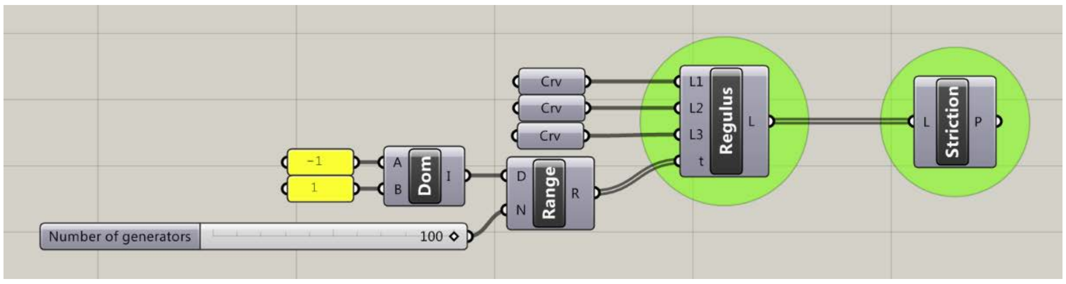

Figure 11. *Regulus in the LineGeometry add-on.*

The fourth component is called **Regulus** (Fig. 11). With it, the ruled surfaces of second order can be generated, whereby the term Regulus emphasizes that only one set of generators is created. Remember, on every ruled surface of second order there are two reguli. Explicitly, this component can generate the hyperbolic paraboloid and the one-sheet hyperboloid (Fig. 12). Three generators are required as input. These are connected individually to the inputs *L1* to *L3* (*Line 1* to *Line 3*), which determines the sequence. At input *t* (*Range of parameters*) a parameter is passed for each desired generator. Somewhat unusual is the parameter range from *-1* to *1*. The first generator at *L1* corresponds with *-1*, the second with *0* and the third with *1*. Here it is again possible to extend the parameter range. However, then it must be reckoned with the fact that the generators are not evenly distributed in the Euclidean sense.

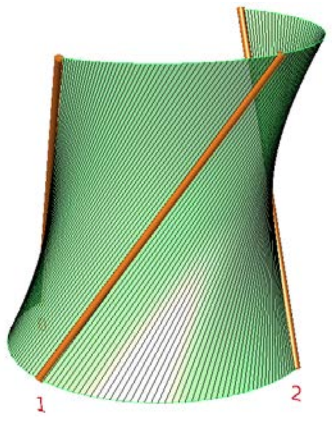

Figure 12. *Regulus through three straight lines (one-sheet hyperboloid).*

2.3. AUXILIARY TOOLS (UTIL)

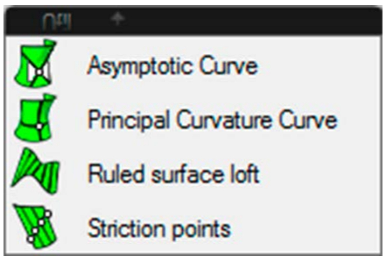

Figure 13. *The Util block in the LineGeometry add-on.*

The auxiliary tools (*Util*, Fig. 13) of LineGeometry are intended to support the work with ruled surfaces and the approximation of freeform surfaces with ruled surfaces.

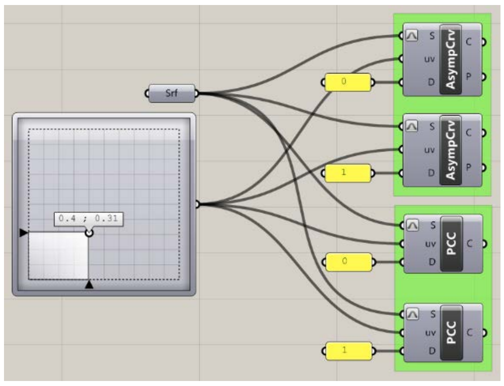

Figure 14. *AsympCrv and PCC in the LineGeometry add-on.*

On non-elliptic surfaces (Gaussian curvature not positive), the first function **Asymptotic curve** (*AsympCrv*) (Fig. 14, upper half) determines the asymptotic curves, i.e. those curves with vanishing normal curvature, passing through a point with certain u-v-coordinates, which are present at the input *uv* (*Point uv*) (Fig. 15). It is useful to activate the input function *Reparameterize* at the input *S* (Surface), so that *u* and *v* can be selected in the domain between *0* and *1*. Since there are generally two asymptotic curves through each point with negative Gaussian curvature, the desired one can be selected at input *D* (*Direction*) with *0* or *1*. The asymptotic curves of a freeform surface are a good orientation for how the surface could be approximated by straight lines, i.e. by patches of ruled surface. For example, the function could be followed by a curvature analysis to determine the "better" (less curved) of the two sets of curves.

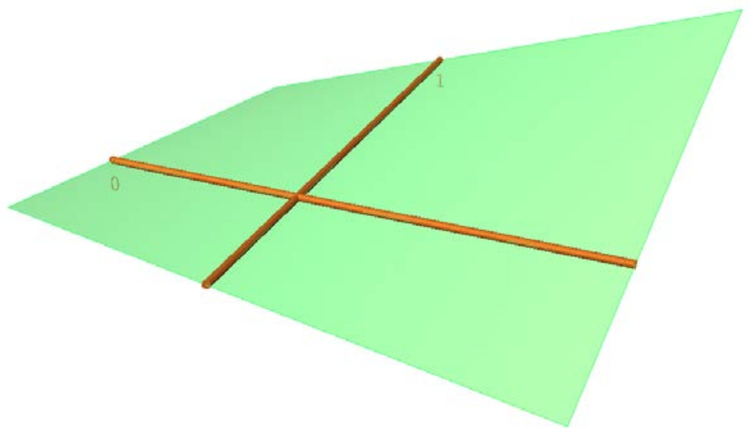

Figure 15. *AsympCrv to a point on a hyperbolic paraboloid yields two generators.*

The second function **Principal Curvature Curve** (*PCC*) (Fig. 14, lower half) determines the principal curvature curves through the point with the coordinates *u* and *v* (Fig. 16). This is of particular interest for the discretization of doubly curved surfaces with planar quadrilateral facets (PQ-meshes) [6].

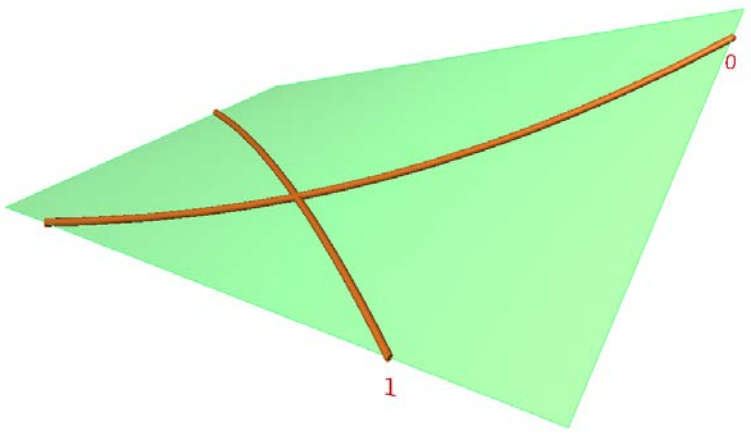

Figure 16. *PCC to a point on a HP surface yields parabolas in the principal curvature directions.*

The third function **Ruled surface loft** (*RuleLoft*) (Figs. 3, 9, 6, 11) is an alternative to Grasshopper's own loft function. The problem with the on-board component is that it becomes very slow when the number of curves is comparatively high, and this is exactly what is useful to approximate the ruled surfaces well. *Ruled surface loft* avoids this problem in a very simple way: the endpoints of the generators are interpolated with curves and then only these two curves are lofted. Since the endpoints determine the parameterization on the curves, the result is identical to the standard loft function - but with much better performance.

The fourth function **Striction points** (*Striction*) calculates the central point on each generator, i.e. the point that is closest to the (infinitesimal) next generator. Striction points are only meaningful on skew surfaces. There they fulfill a significant curve, the striction curve, along which the ruled surface has its strongest curvature. The curve is also relevant from a structural point of view, since the highest stiffness can be expected there (Fig. 17). The striction points are calculated approximately and are more accurate the more generators in *L* (*Generators of the ruled surface*) are passed. Striction points can also lie outside of the represented surface regions.

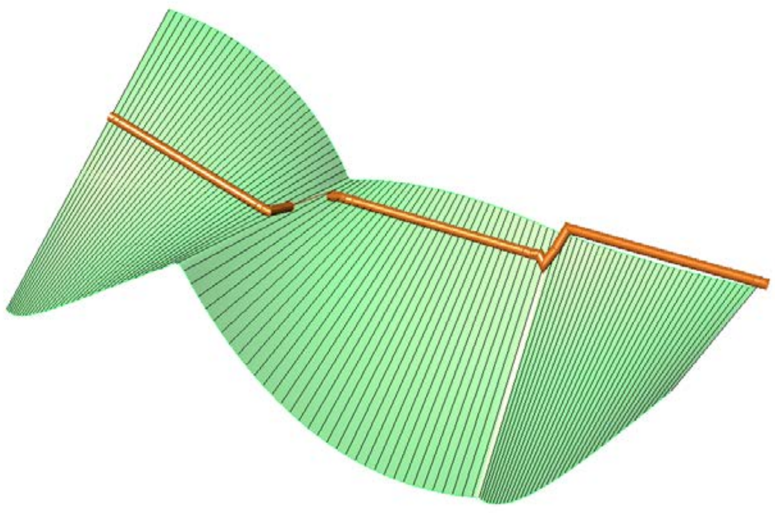

Figure 17. *Striction curve of the surface of Fig. 08, which, contrary to the illustration, is not continuous. The three discrete striction lines of the three helicoidal segments coincide inevitably with their respective axes.*

3. APPLICATIONS AND CONCLUDING REMARKS

The add-on *LineGeometry* was used in the development of the demonstrator "Shell Bridge with Ruled Surface Geometry" in SPP 1542, funded by the German Research Foundation (DFG) [7]. There, line geometric modeling was combined with finite element methods in an optimization process. Furthermore, the add-on was the starting point of projects during the summer school "Line Geometry for Lightweight Structures" in September 2018 at TU Dresden [8]. In these projects, the add-on was particularly convincing in that the mathematical superstructure makes the ruled surfaces appear more elegant and smoother than could be achieved with the usual methods. In this respect, it pays off that the interpolation algorithms are in a natural way based on the helicoid, i.e. a minimal surface.

ACKNOWLEDGEMENTS

The add-on LineGeometry was developed under the supervision of the author by the research group *Geometric Modeling and Visualization* (GMV) at the Institute of Geometry, TU Dresden, in the course of the priority program "Concrete Light" (SPP 1542), funded by the German Research Foundation (DFG). Significant contributions were made by Markus Hagemann, Daniel Klawitter and Michael Pott. Martin Eichenauer (also GMV) produced the formwork elements for the shell bridge demonstrator in the *Makerspace* of the SLUB Dresden. The project partners of the TU Berlin from the *Chair of Conceptual and Structural Design* lead by Mike Schlaich were responsible for the production of the shell bridge and the software interface to SOFiSTiK. The interface, named GHtoSOFi, was programmed by Konrad Freymann and Juan Pablo Osman Letelier (prize winners in the competition *Auf IT gebaut* 2018). My sincere thanks go to all those involved and to the DFG.

LITERATURE

[1] D. Lordick, D. Klawitter, M. Hagemann, "Liniengeometrie für den Leichtbau," in Leicht Bauen mit Beton. Forschung im Schwerpunktprogramm 1542, Förderphase 1. S. Scheerer,
[2] M. Curbach, Ed. Dresden: Institut für Massivbau, Technische Universität Dresden, 2014, pp. 224-235.
[3] H. Pottmann, J. Wallner, Computational Line Geometry. Heidelberg, Berlin etc.: Springer, 2001.
[4] M. Hagemann, D. Klawitter, D, Lordick, "Force Driven Ruled Surfaces," in Journal for Geometry and Graphics (JGG), vol. 17, No. 2, pp. 193-204, 2013.
[5] Daniel Lordick. "Add-on for Grasshopper in Rhinoceros 3D – LineGeometry". Internet: http://linegeometry.daniellordick.de, [May 2, 2022].
[6] D. Lordick, "Intuitive Design and Meshing of Non-Developable Ruled Surfaces," in Proceedings of the Design Modelling Symposium Berlin, 2009, pp. 248-261.

[7] J. P. Osman Letelier, A. Goldack, M. Schlaich, D. Lordick, J. Grave: "Shape optimization of concrete shells with ruled surface geometry using line geometry," in Proceedings of the IASS Annual Symposium 2017 »Interfaces: architecture.engineering.science«, 2017, Hamburg, Germany, 2017, Paper #9199 1-10 (published on a USB flash drive).

[8] C. Ress, D. Lordick. "[Documentation of the] Summer School LGLS at the TU Dresden – long version". Internet: https://www.youtube.com/watch?v=6ZdpHvWthRI, [May 2, 2022].