Designing for Hot-Blade Cutting

Geometric Approaches for High-Speed Manufacturing of Doubly-Curved Architectural Surfaces

David Brander, Andreas Bærentzen, Kenn Clausen, Ann-Sofie Fisker, Jens Gravesen, Morten N. Lund, Toke B. Nørbjerg, Kasper Steenstrup, and Asbjørn Søndergaard

D. Brander, A. Bærentzen, A. Fisker, J. Gravesen, T. B. Nørbjerg, K. Steenstrup
Technical University of Denmark, Denmark

dbra@dtu.dk
janba@dtu.dk
ansofi@dtu.dk
jgra@dtu.dk
tono@dtu.dk
khor@dtu.dk

K. Clausen
3XN Architects, Denmark

kec@3xn.dk

A. Søndergaard
Odico Formwork Robotics, Denmark

asbjorn.sondergaard@aarch.dk

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

Abstract

In this paper we present a novel method for the generation of doubly-curved, architectural design surfaces using swept Euler elastica and cubic splines. The method enables a direct design to production workflow with robotic hot-blade cutting, a novel robotic fabrication method under development by authors of the paper, which facilitates high-speed production of doubly-curved foam moulds. Complementary to design rationalisation, in which arbitrary surfaces are translated to hot-blade-cuttable geometries, the presented method enables architects and designers to design directly with the non-trivial constraints of blade-cutting in a bottom-up fashion, enabling an exploration of the unique architectural potential of this fabrication approach. The method is implemented as prototype design tools in MatLAB, C++, GhPython, and Python and demonstrated through cutting of expanded polystyrene foam design examples.

Keywords:
robotic fabrication, hot blade, digital design, EPS-moulds, cost-efficiency, concrete structures

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

Figure 1. Louisiana State Museum and Sports Hall of Fame, courtesy Trahan Architects (top). Kagamigahara Crematorium, Courtesy Toyo Ito Architects (bottom).
Copyright, Figure 1a: Trahan Architects. Copyright, Figure 1b: Toyo Ito Architects.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

1. Introduction

In contemporary architectural practice, a rising number of projects employ advanced building geometries, which departs from the orthogonality of mainstream construction, incorporating digital design tools and manufacturing for the realisation of expressive or dynamic design features (Pottman 2007). A group of projects within this category, such as Kagamigahara Crematorium (Toyo Ito Architects, 2006) and Waalbridge Extension (Zwart & Jansma, 2015), rely on the doubly-curved geometries, which may be constructed either via production of manual formwork, which relies on digitally produced guides to bend plate material in place over large radii. Alternatively, large-scale CNC-milling of either foam molds for concrete casting or direct milling of construction materials are employed enabling the realisation of shorter radii designs with more detail and surface controls. Such projects include, for example, Spencer Dock Bridge (Amanda Levete Architects, 2010), Louisiana State Museum and Sports Hall of Fame (Trahan Architects 2013), Museum Foundation Louis Vuitton by Gehry & Associates (Paris, 2014); the Nordpark cable railway by Zaha Hadid Architects (Nordpark 2007), the Metz Pompidou by Shigeru Ban (Metz 2010).

However, none of these general construction processes provides a cost-effective option for general construction, and projects of this type therefore require extraordinary budget frameworks for realisation: Manual onsite formwork processing in this category is a highly laborious and demanding process, with resulting difficulties in cost-engineering to follow. Large-scale CNC-milling on the other hand, provides cost transparency due to the digital nature of the process – although the mechanical principle of CNC-milling, which subtracts material through incremental removal, is inherently slow when applied to architectural scale production, and results in exuberant machining times and high costs.

Recent developments in architectural robotics and digital manufacturing have seen the emergence of a number of approaches to diversify the machining options available, with the purpose of realizing structures of more advanced geometries. This includes actuation of a flexible membrane as a casting surface (Jepsen et al. 2011; Hesse 2012); dynamic slip-casting for column elements (Lloret et al. 2014), as a variant of the additive manufacturing of concrete structures (Khosnevic 1998, Lim et al 2012); fabric formwork applied as an alternative technique for the casting of advanced designs (Veenendaal et. al 2011); spatial wire cutting (Rust et al. 2015) as well as large-scale robotic hot-wire cutting of EPS molds by authors of this paper. None of these approaches however, are capable of delivering combined (1) unconstrained degrees of freedom which enables general purpose realisation equivalent to that of CNC-milling; (2) machining efficiencies which significantly supersedes that of CNC-milling; (3) process predictability which ensures the delivery of a pre-controlled geometry.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

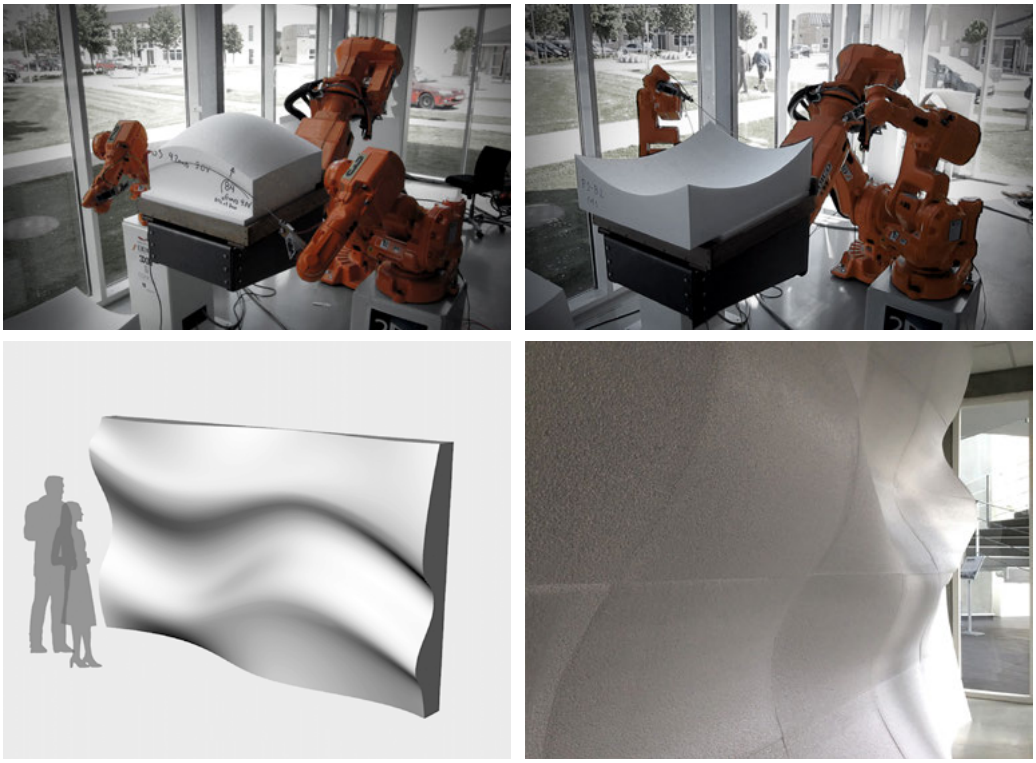


Figure 2. Bladecutting experiments in progress.
Top: 18-axis tri-robot hot-blade pilot-cell.
Below: concrete panel design and cut foam result.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

2. Robotic Hot-Blade Cutting

In an effort to develop a new manufacturing process which would satisfy these criteria, the authors of this paper initiated in 2012 the Bladerunner project, which targets the cost-effective production of double-curved foam moulds. The technique developed in this effort – dubbed robotic hot-blade cutting – employs a multi-robotic process, in which an 18-axis cell consisting of 3 industrial manipulators translates a flexible, heated blade through expanded polystyrene blocks in a thermal cutting process, while controlling the distance and rotation of end-effectors to achieve variable cross-section curves along the trajectory of cutting sequences. Pilot production experiments currently under development seek to explore and demonstrate the applicability of this method for production of pre-fabricated concrete elements under a general CAM paradigm, in which arbitrary design input – understood here as geometry which is conceived without particular regard to the specific constraints of the process – is rationalised for hot-blade production using a set of algorithms developed within the project. These early developments point to the perspective of a highly time-efficient production method, up to 126 times faster than comparable CNC. However, complementary to a top-down process of rationalisation, a second trajectory is also possible, in which the geometric constraints of the hot-blade cutting is incorporated already in the design process, thus operating under a generative design paradigm. The work in the following chapters outlines tools and processes that can facilitate such an approach.

3. Designing with Elastica

An Euler elastica is the shape assumed by an elastic rod with planar constraints of position and tangents placed only on its endpoints. A planar curve is geometrically determined by an angle function $\theta(t)$, the angle between the tangent and some fixed direction. The angle function for the elastica are given by solutions of the normalised pendulum equation $\theta = -\sin\theta$, a nonlinear equation the solutions of which can fortunately be given explicitly in terms of elliptic functions.

Mathematically, the correct model for an elastica was given by James Bernoulli in 1691 (see Truesdel 1983). He approximated the solution for the case that the ends of the rod are perpendicular to each other, recognizing that non-standard functions were needed for an analytic expression. The problem was subsequently suggested to Leonhard Euler in 1743, who gave all possible shapes for the elastica in his famous treatise on the calculus of variations (Euler 1744).

People have in fact been designing with elastica for centuries, albeit in a physical rather than mathematical format. Prior to the introduction of computers for draughting in the shipping, aviation, and automobile industries, which began in the

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

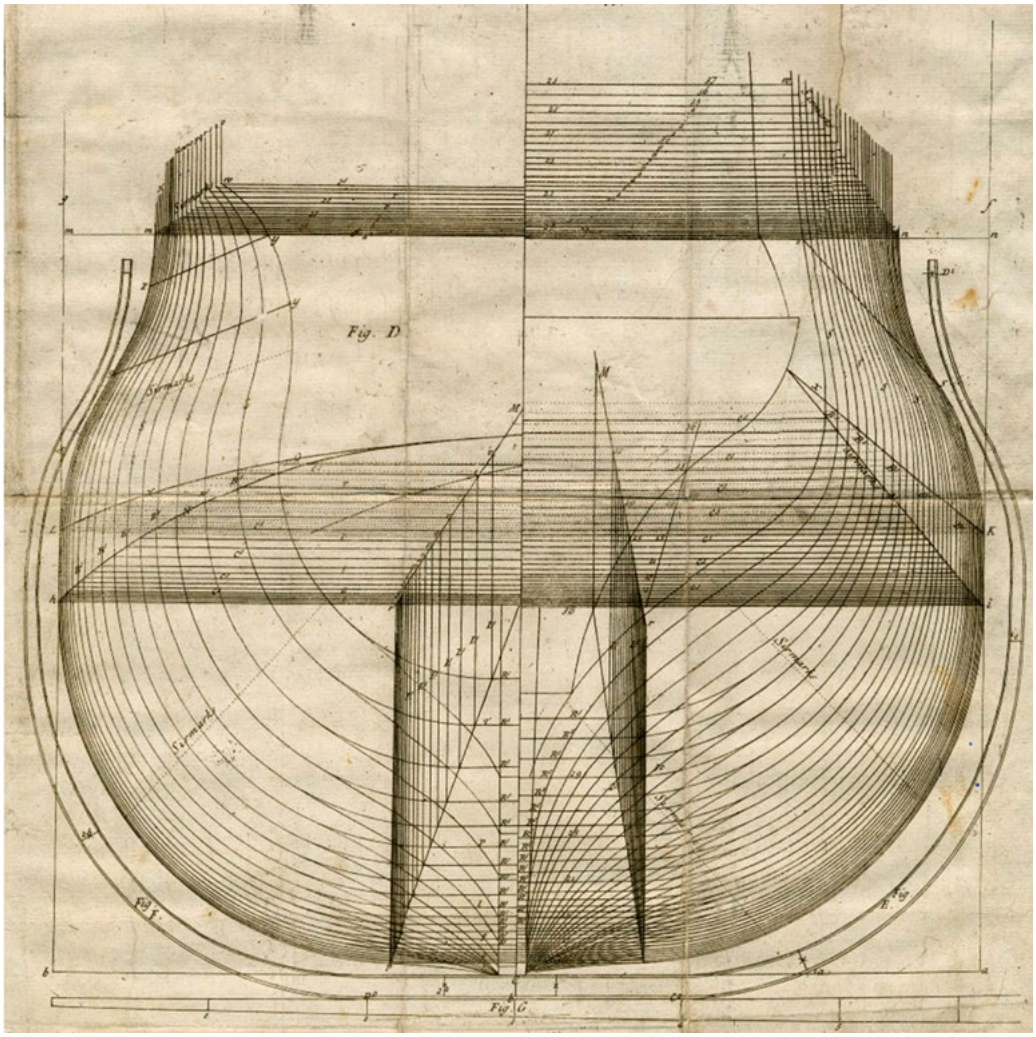


Figure 3. Use of physical splines for ship-hull manufacturing.
Copyright: William Sutherland, The Shipbuilders Assistant: or, Some Essays Towards Completing the Art of Marine Architecture (London, 1711).

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

1950s, the curves needed in the designs were created by tracing the shapes of thin wooden rods, known as splines, manipulated by the placement of so-called 'ducks' at various points to create a naturally smooth curve. This practice started in the ship-building industry, where the placement of the ducks simulates the placement of ribs in the hull of the ship; hence the curve drawn by following the spline is an accurate reflection of the natural shape adopted by the planks forming the ship's hull. The drawing took place at the loft of the shipyard, hence the word 'lofting', now used in the CAD industry. Going further back in time, splines were used for the storage and transmission of designs in ancient Rome, in the form of physical templates for the ribs of ships (see Farin 2002).

When computers became cheaper and more powerful, a desire for electronic storage and editing appeared. The word 'spline' now began to be used for piecewise polynomial or rational curves used in design. Paul de Casteljau at Citroën and Pierre Bézier at Renault used what are now known as Bézier curves to describe the designs. In the USA, Carl de Boor at GM used B-splines (basis splines) for the designs. In the aircraft industry, at Boeing, similar developments took place.

3.1 Design vs. Rationalisation for Hot-Blade Cutting

For a CAD surface to be produced using hot-blade cutting, it needs to be segmented into suitable pieces and each surface segment then swept by planar curves that are subsequently approximated by elastic curve segments. We described this rationalisation process in recent and forthcoming work.

An alternative to the rationalisation of a CAD design is to provide design tools that allow designers to create fabrication-ready surfaces. There are a number of reasons for doing this: Firstly, the rationalisation of an arbitrary design is non-trivial and in general can result in some regions of the surface needing to be produced by another method such as milling. Secondly, a design tool can give the designer control over the cast-lines between the segments, which will in many cases be visible from close range.

A third reason is the additional complexity arising when we consider a surface created by more than one cut to the same EPS block. For example, consider the surface shown on the left in Figure 4. By cutting the same EPS block twice, the second time with a 90 degree rotation, the surface on the left in Figure 5 is produced. Now the surface on right in Figure 4 approximates the first surface very well at the end-points of the cutting blade, but deviates slightly in the middle. Such an approximation is likely to arise in surface rationalisation, because we will usually need the patches to fit together with tangent continuity, which requires a little more freedom away from the patch edges. Doing the same two cuts with the new surface results in the surface on the right in Figure 5, and here we can see that the intersection curves are no longer the same as the design.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

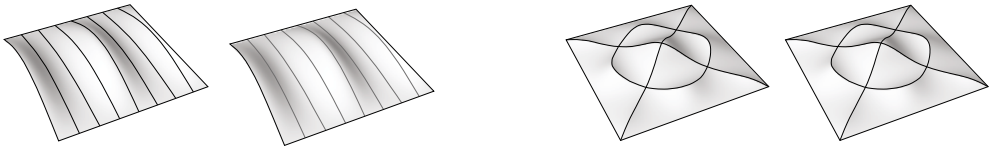


Figure 4 & 5. (right) Matlab generated surface. (left) rotated double cut of the same design.

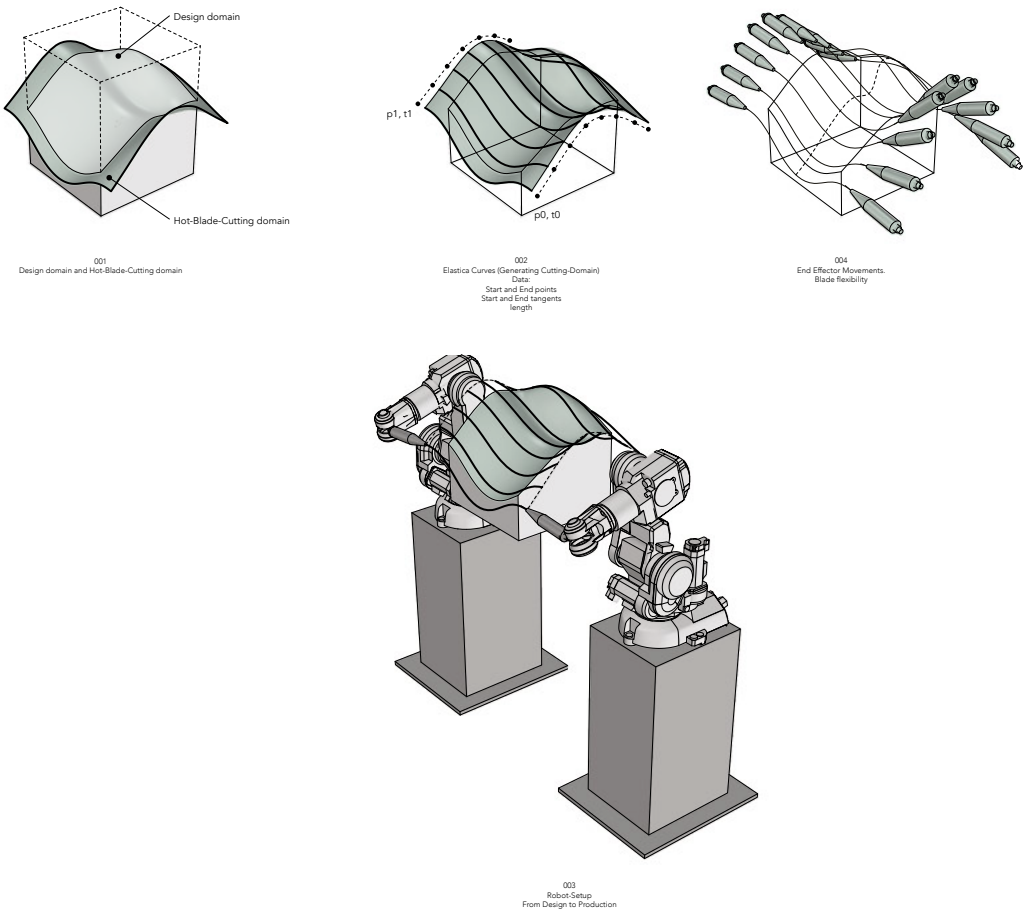


Figure 6. Configuration of input data.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

Of course there are solutions for this kind of issue in the rationalisation approach, but this scenario illustrates the kind of advantages one has with a fabrication-ready design tool.

3.2 Single Block Designs

For the simple case of just one block, we design with curves of the desired length, i.e. the length of the cutting tool. During cutting the cutting tool is kept in a horizontal plane perpendicular to the cutting direction. The data needed for the robot movement are thus simply the positions of the ends of the blade and the angles of its ends relative to horizontal. The positions are given as $y, z\text{-}coordinates$ (the $x\text{-}coordinate$ describes how far the curve is in the cutting direction), see Figure 6. In the design space (e.g. Rhino) we know the position of the design curves relative to the EPS block, so we can obtain the robot data directly from the design curves by computing the positions of the endpoints relative to the EPS block, and their angles relative to horizontal.

The plugin for the discrete elastica ensures that we get a representation of the final design in Rhino, before going to production. In the following, we describe the numerical algorithm used to find this solution given end points, tangents at the end point, and the length of the curve (see Bruckstein et al. 2001). This method does not find the elastica in terms of the elliptic functions; instead we return to the defining property of elastic curves, namely that they minimize $\int \kappa^2 ds$. In a discrete setting where we represent the curve using line segments of equal length, the analogous energy for the piecewise linear curve is:

$$F_a(\alpha) = \sum_{i=0}^{n-1} \alpha_i^2$$

where α^i is the turning angle at segment i as illustrated in Figure 7.

Thus, to find the discrete curve, we minimize F_a subject to two constraints that both serve to enforce the boundary conditions. To ensure that the tangent at the last point has the correct direction, we require that

$$0 = F_t(\alpha) = \sum_{i=0}^{n-1} \alpha_i - (\alpha_n - \alpha_{-1})$$

where α_{-1} and α_n are the angles that correspond to the direction of the first and last line segment, respectively. Clearly, we also require that the curve ends at the right point. This is taken care of by the second energy

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

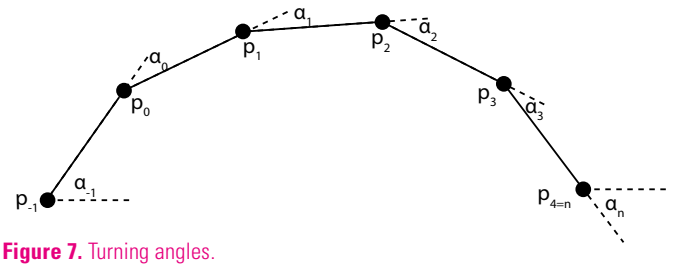


Figure 7. Turning angles.

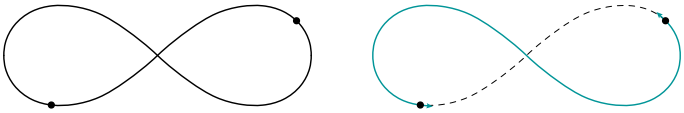


Figure 8. Turning angles for a discrete elastic.

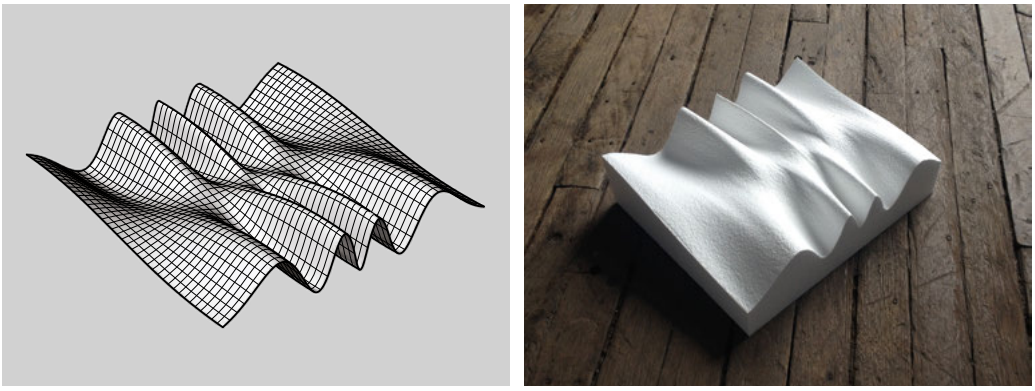


Figure 9. Examples of a generated and a cut surface.

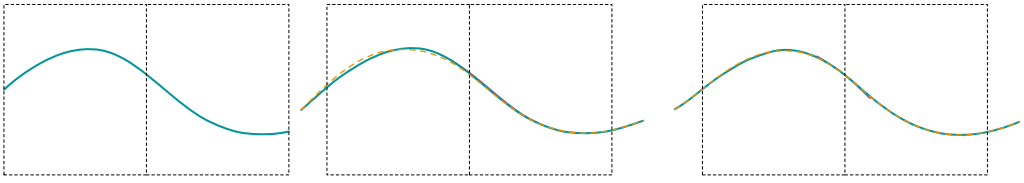


Figure 10. In blue an exact elastic curve and in black a discrete approximation calculated from the boundary conditions shown in yellow.

Figure 11. Transitions between elastica.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

$$F_{p(\alpha)} = \left\| \sum_{j=-1}^{n-1} \left[\cos\left(\sum_{k=-1}^{j} \alpha\_k \right), \sin\left(\sum_{k=-1}^{j} \alpha_k \right) \right]^{\mathrm{T}} - (p_n - p_{-1}) \right\|$$

We can construe the discrete elastic curve problem as an inverse kinematics problem. In this light, F_p simply ensures that the end of the curve (end effector) coincides with p_n – the end point of the elastic curve.

In practice, we find the elastic curves using a two-step procedure. In an outer loop, we reduce F_a and in the inner loop, we solve for $F_p{=}0$ and $F_t{=}0$. F_a and F_p are minimised using gradient descent, whereas F_t can be solved exactly in each step. The outer loop runs for a fixed number of iterations whereas the inner loop terminates when a desired tolerance has been reached. Figure 11 shows a test. The green figure-eight is an analytic elastic curve. The black curve on top is the discrete approximation computed from the boundary conditions shown in red.

3.3 Examples

The images below show a surface defined as the graph of a bivariate function

$$z = f(x,y) = k \cos\left(2\left(\frac{\pi}{2} - y\right)^4\right) \sin(x) \exp(-0.1x^2) \quad [x,y] \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \times (-0.75, 0.75)$$

z = f(x,y) =k(2(/2-y)4)(x)(-0.1 x2) (x,y) (-/2, /2)(-0.75,0.75)

As the initial design created using a Python script in Rhino, as a rationalized surface and as a styrofoam block cut with the hot blade.

3.4 Design Using Multiple Blocks

For the more complex case with several blocks, a more advanced procedure is used to ensure a smooth transition from one block to the next. Consider a curve design that passed over two blocks (see Fig. 10, left). We need to produce this in two cuts – one per block – and we want the two block segments to match at the boundary after cutting.

If we run our plugin independently on the two parts of the curve, we would automatically obtain the smooth transition, but we would be unable to get the necessary data for the robots, since the curves (which are inside the blocks) are shorter than the cutting tool, and we have no quick way to extend these while preserving the elastic properties. If we simply extend the original curve and then use our plugin on parts with the desired length, we do not ensure smooth transition (see Fig. 10, right).

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

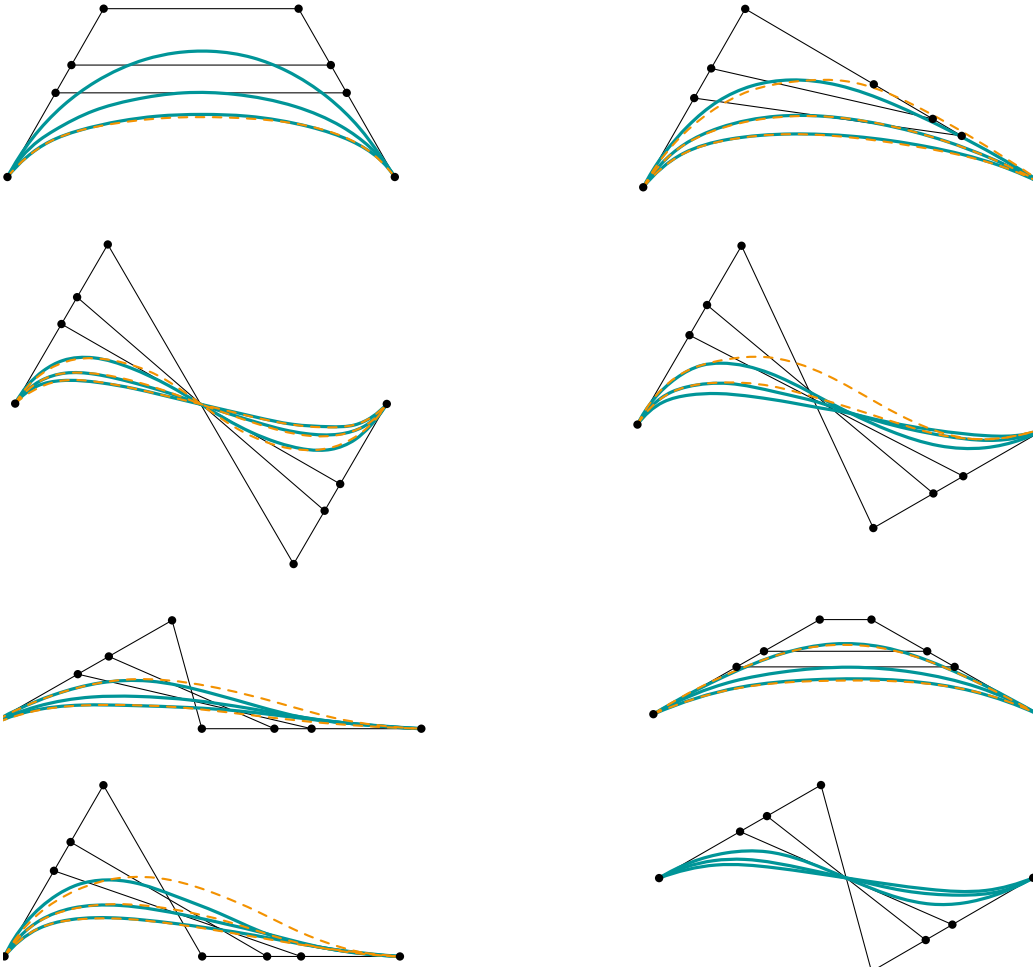


Figure 12. In blue Bézier curves, in black their control polygons, and in dashed red elastic curves with the same endpoints, end tangents, and lengths.

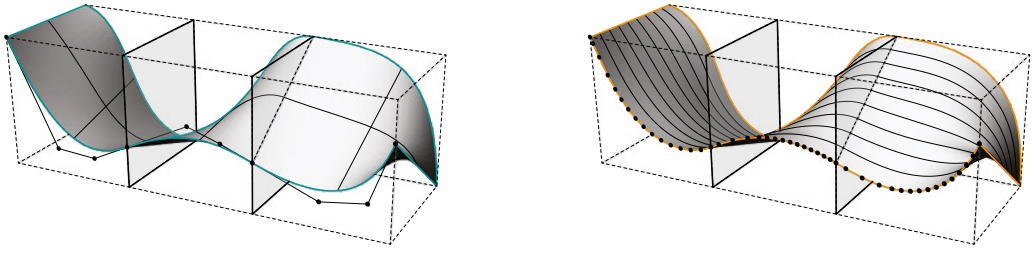


Figure 13. A Rhino design tool. Top: three blocks with two tangent continuous curves. Lower left: lofted Bézier curve surface. Lower right: rationalized elastica curve surface.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

The solution is, instead of finding a discrete representation of the elastic curve that models the shape of the cutting tool, we find an analytic mathematical description of the elastic curve in terms of elliptic functions. This requires a more cumbersome optimisation in order to find the parameters that describe the rationalised curve. However, when these parameters are found, the entire (infinitely long) elastic curve that contains the required segment is known. It is then simple to extend the segment to an elastic segment of the length of the blade (see Fig. 11) and from this extract the position data for the robots.

4. An Alternative Approach: Bézier Curves as a Proxy for Elastic

Historically, the use of cubic splines as a design tool was often motivated by saying that they are a good substitute for real physical splines. This is justified by the fact that, if the speed of a curve is constant, then the square of the curvature is the same as the square of the second derivative, and if the latter is minimised we obtain exactly a cubic spline (see Yamaguchi 1988). Now a cubic curve does not have constant speed unless it is a straight line; but if the control polygon of a cubic Bézier curve is reasonably well behaved, then the curve is close to an elastica. See Figure 12, where 24 Bézier curves are plotted together with elastic curves having the same endpoints, end tangents, and length.

If the angles in the control polygon are not too acute, then there is very little difference between the Bézier curve and the elastic curve of the same length and end conditions.

Based on this observation, we implemented a design tool in Rhino™ where the surfaces and their rationalisation are very close. The idea is that we imagine space filled with EPS-blocks and define our surfaces such that the parameter curves have exactly one planar cubic piece in each block. As a simple example consider Figure 13, where we have three blocks and have defined three Bézier curves at both the front and the back of the blocks in such a way that they have common endpoints and tangents, i.e. they form two tangent continuous curves (see Fig.14). The surface is defined by a lofting process. We can of course have several layers of blocks, and we could also have included more curves in the middle of the blocks. The inputs are planar curves with exactly one cubic piece in each block, and the surface is defined by lofting.

One can achieve a curvature continuous construction by replacing the sequence of Bézier curves by a single planar cubic spline (with simple interior knots). Between the knots we have a cubic polynomial, so we if we require that the image of the knots is on the block boundaries then we obtain a single cubic polynomial piece in each block (Fig. 15).

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

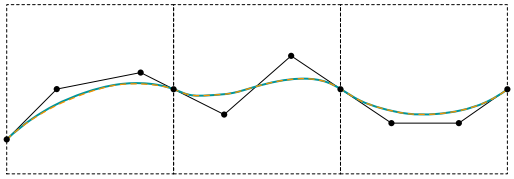


Figure 14. The tangent continuous construction. In each block we have a cubic Bézier curve and we require that adjacent curves have control polygons the first and last legs of which form a single line segment. In dashed red we have plotted the true elastica having the same length and boundary conditions as the Bézier curves.

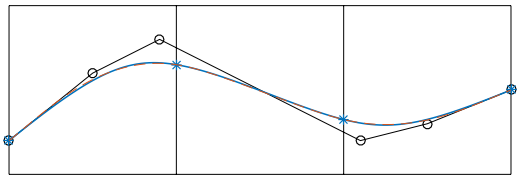


Figure 15. The curvature continuous construction. A single cubic spline curve where the images of the knots are on the block boundaries. In dashed red we have plotted the true elastic curves that have the same length and endpoints as the polynomial pieces. The tangents corresponding to the extreme points of the cubic curve are also prescribed.

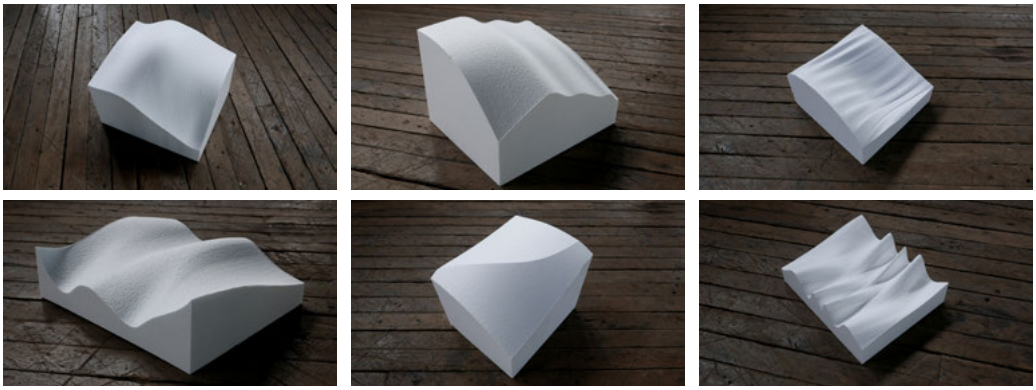


Figure 16. pre-test cuts from the workshop. From the left: while single, continuously swept surfaces are readily achievable through rationalization, the ripple and curvature effect on the right most samples requires careful alignment with the blade directions, and hence is difficult to obtain aside from directly controlling it in an elastica-swept surface.

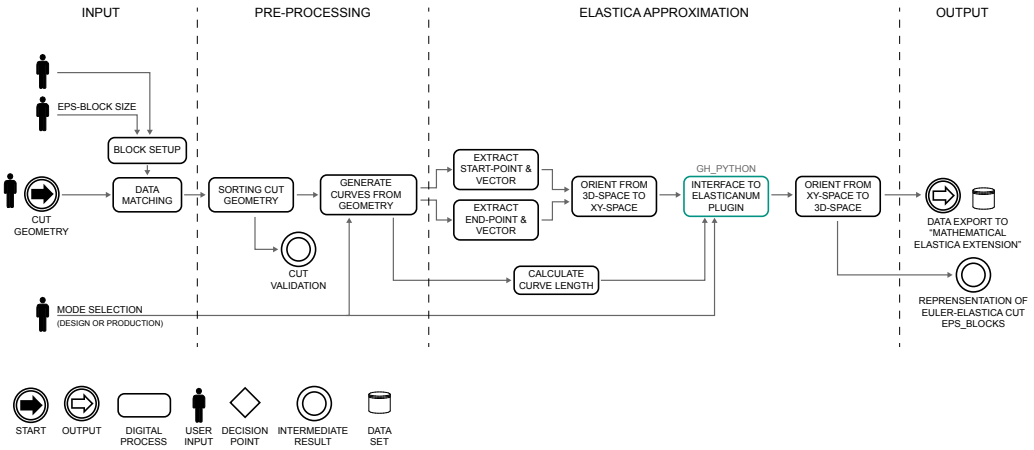


Figure 17. Data workflow chart.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

If we replace the spline with an elastica having the same length in each block as the spline, passing through the images of the knots, and having the same tangent angles in the beginning and end as the spline curve, we obtain almost the same result. This corresponds exactly to the classical design using a physical spline and ducks. We just have to place the ducks at the images of the knots.

5. GH Workflow

For the development of design experiments as well as participatory workshop design sessions, a toolset is developed in McNeel Grasshopper, implementing the above approaches. The toolset is linking the full cycle of research, innovation, implementation and production, creating a framework for geometric operations consistent with the robotic setup. The overall logic of the workflow connects conventional digital modeling approaches with the robotic hot-blade process. This requires the identification and rationalisation of geometry types before rebuilding the geometry to the accuracy of the robot, EPS-segmentation and tolerances.

The Grasshopper-definition is developed with the purpose of designing with rationalisation through Euler elastica. It is a Real-Time process that allows for fast interpolations from design to production and ensures a smooth curve continuity transition from one block to the next. The tool is very flexible and allows for large variation of form typologies when designing with single or multiple cuts in the design explorations.

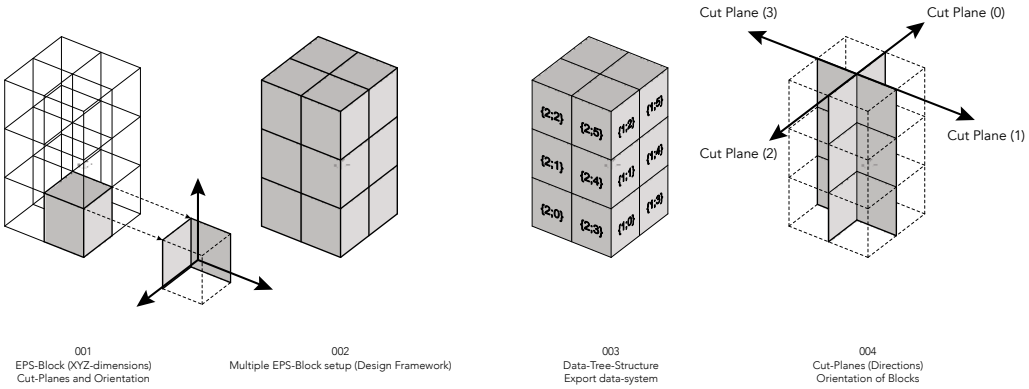
The setup is part of a larger digitised workflow; *'Interpolation of Geometry'*, *'Euler Elastica Approximation'*, *'Mathematical Elastica Extension'*, which is is a linear process that allows for feedback loops when data or geometry are outside of preset conditions and/or needs changes. The Grasshopper tool *'Design with Elastica'* inputs arbitrary surfaces and/or curves, converts them into planar elastica curves that describe the cut-direction and the movement of the robot-setup. The setup is structured in four overall processes; *'Global Parameters'*, *'Input'*, *'Process (Machine)'*, *'Output (Export)'*.

5.1 Global Parameters

The backbone of the setup is the global parameters that are changeable within the workflow environment. Its settings, syntax, and data are adapted in both the approximation plugin and the extension script while the workflow recalibrates when global settings are re-configured or need additional inputs. The global parameters allow you to toggle between *'Design'* or *'Production'* which are value bases and changes the resolution of the Elastica approximation.

One block is locally defined by its XYZ-values (dimensions), cut-plane (orientation) and its local location in the XYZ-world-coordinate system and global

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html



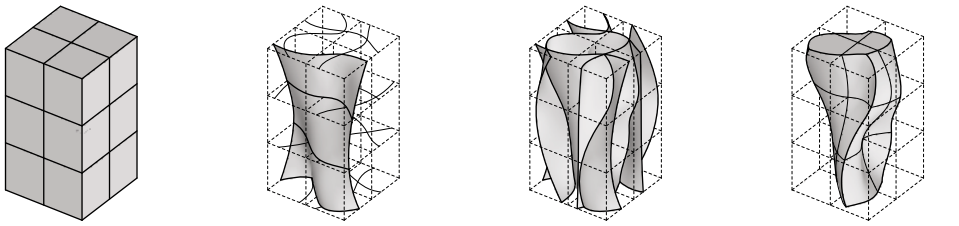
001
EPS-Block (XYZ-dimensions)
Cut-Planes and Orientation

002
Multiple EPS-Block setup (Design Framework)

003
Data-Tree-Structure
Export data-system

004
Cut-Planes (Directions)
Orientation of Blocks

Figure 18. Work object configuration.



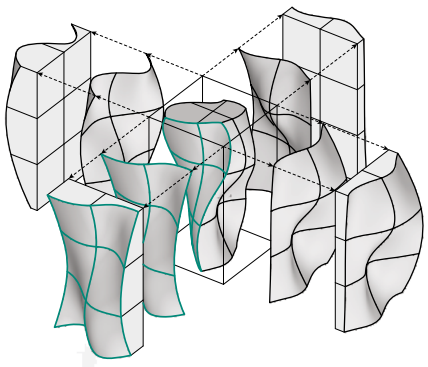
005
Design Framework

006
Geometry for cutting blocks (Design input)

007
Multiple Design "Inputs"
Data-Tree-Structure matches Design Framework

008
Brep represntation of 3Dimensional design

Figure 19. Generation of elastica design surfaces.



009
Sorting Cut-Geometries
Location and rotation of EPS-blocks in 3D space and block X and Y domain in the plane

Figure 20. Configuration of demonstrator design.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

location in a multiple-block-system. The block and blade length are interconnected to each other, if too short, the robot will move into the EPS-block, while too large the physical implications will increase and affect the precision.

5.2 Input

A multiple block configuration is developed as a framework for the generation of continuous surfaces over several blocks. The *'block setup'* is framed in a data-tree structure that matches and sorts the input designs for each block and subsequently for each face of the block (6 sides). The procedure generates a data-list for each block containing cut-plane, cut-direction, number of cuts, rotation, and location. By defining a clear data-flow from the input step you gain full control from design intent till export code and production.

5.3 Preprocessing of Geometry

The pre-processing step first matches each block (nested in a data-tree-structure representing the design framework and block number within the design framework) with cutting geometries related to the blocks design framework. The cutting geometry is then sorted according to cut priority, the primary cut being closest to the blocks base, and the remaining cuts are checked for collision with the primary cut and removed if no collision occurs. The final operation generates a number of planar curves for each cut by sampling the cut geometries in the X direction of the blocks. The sampling is extended beyond the block domain to allow the robots to have lead-in and lead-out of each block. The number of sampling points is triggered by the current mode selected (design or production).

5.4 Elastica Approximation

To approximate the Nurbs curves we made a Rhino plugin. The *ElasticaNum Plugin* operates on curve start-, end-points, tangent vectors at start- and end-points, and the desired length of the elastica curve. The *ElasticaNum Plugin* is interfaced through the Rhino command-line and python code using a custom data structure. The final operation in the *Elastica approximation* is reorienting the curves back into 3D space.

5.5 Output

The *'Design with Elastica'* tool generates two outputs that work parallel within the workflow and connect the designer with the final rationalisation output while still designing in the beginning of the process. Output one is data-driven

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

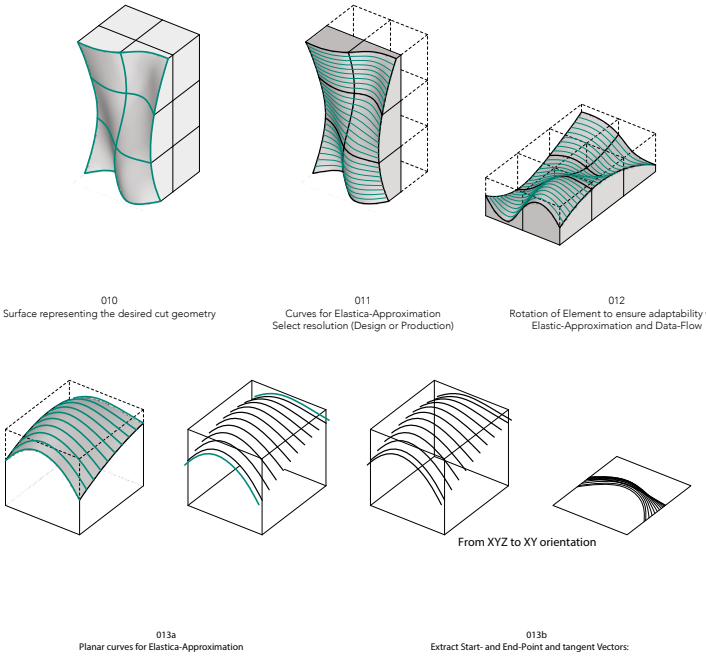


Figure 21. Description of production surfaces via swept elastica cross sections.

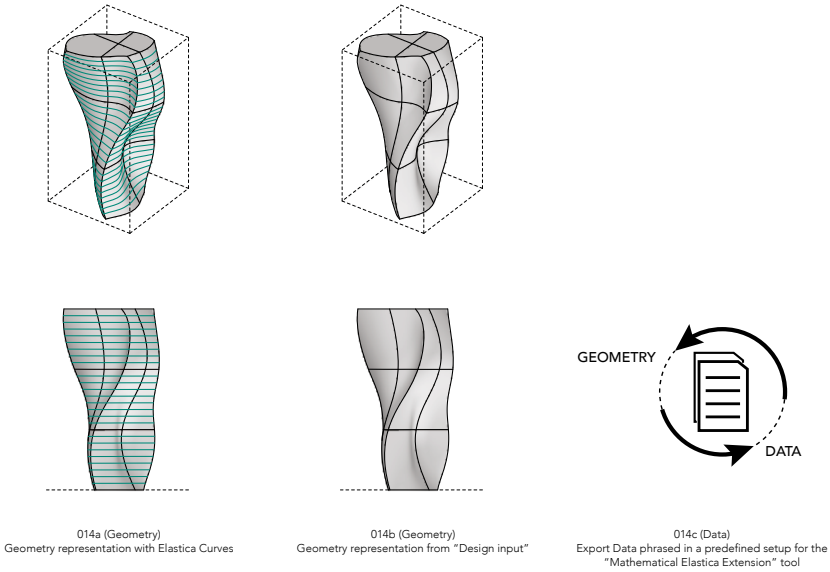


Figure 22. Global orientation of constitutive elastica cross-sections.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

exports to the *'Mathematical Elastica Extension'*, while output two is representing the desired geometry by Elastica Curves. By defining a data-structure a mutual adaptation of data import and export was defined from Input, Process and Output. The digitised workflow outputs data from the *'Design with Elastica'* tool to the *'Mathematical Elastica Extension'* and weave the tools together. To ensure a smooth transition from one block boundary to the next, the necessity for data conversion is important to perform a mathematical extension that preserve the elastica properties.

The setup allows for a real-time *design to fabrication workflow* and a comparison between the *Design-Geometry* and the *Elastica-Geometry* are processed. While the Design-Geometry is an arbitrary input, the *'Euler Elastica Approximation'* curves are lofted with the setting on *tight*, which uses square root of chord length parameterisation in the loft direction.

5.6 Design Workshop

The developed toolset was subsequently tested in a workshop setting with 16 participants in the format of the Superform: Robotic Hot-Blade Cutting workshop, held March 15–18 2016 in extension to the Robarch 2016 conference at Walsh Bay, Sydney. The workshop tasked participants with formal explorations of hot-blade design potentials, produced through a dual robot setup consisting of 2x ABB IRB 1600 manipulators in a MultiMove configuration. The explorations uncovered several benefits of working directly in a production-ready geometry: Firstly, the exploitation of double (or more) cuts, in which two intersecting surfaces creates a sharp crease is a feature difficult to approximate through rationalisation (Fig. 13, second row, middle). Secondly, the design of expressive ripple or wave-effects (Fig. 14) requires careful alignment with blade-cutting direction and curvature description to remain feasible. As such, they exemplify design potentials difficult to achieve through linear rationalisation.

6. Conclusion

A set of methods has been proposed for design generation of surfaces which incorporates the constraints of an elastic blade swept mechanically by two or more industrial robot manipulators. The methods are implemented as prototype design tools in C++, MatLAB, Python, and GhPython to enable interaction with non-specialist designers. The toolset was tested with 16 participants in the RobArch 2016 workshop: Superform – robotic hotblade cutting. The workshop design experiments revealed several design features that would be difficult to achieve in pure rationalisation workflows, as a result of the direct incorporation of constraints and live design feedback enabled by the framework.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

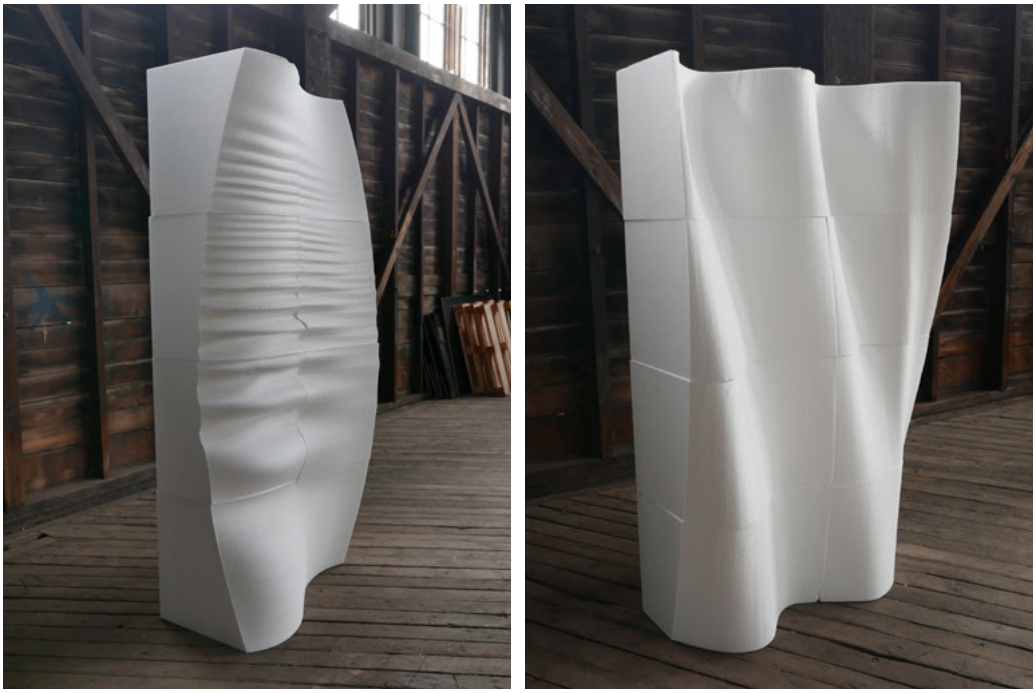


Figure 23. Examples of the workshop participants design explorations. Design left: Jill Smith & Phil Dench. Design, right: Dharman Gersch.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

Acknowledgements

The work presented in this paper was developed within two 3-year research projects, 'BladeRunner (2013-16)' and 'Digital Factory (2015-18)' supported by the Innovation Fund Denmark, as part of a larger effort in development of robotic hot-blade cutting manufacturing methods. The BladeRunner project is developed by partners Odico Formwork Robotics Aps (project lead); the Technical University of Denmark, Department of Mathematics and Computer Science and Department of Mechanical Engineering; the Danish Institute of Technology; 3XN Architects / GXN Innovation and Confac A/S. Digital Factory is developed by partners Odico Formwork Robotics Aps (project lead), the Technical University of Denmark, Department of Mathematics and Computer Science and 3XN Architects / GXN innovation. The authors would like to thank the participants of the Robarch Superform workshop for their dedicated experimentations during Robarch 2016, Sydney.

References

Brander, D., J. Gravesen, and T. Nørbjerg. Approximation by planar elastic curves. arXiv:1509.00703[math.NA].

Bruckstein, A.M., R.J. Holt, and A.N. Netravali. 2001. "Discrete Elastica." *Applicable Analysis* 78, 3–4: 453–485.

Bruckstein, A.M., A.N. Netravali, and T.J. Richardson. 2001. "Epi-Convergence of Discrete Elastica." *Applicable Analysis* 79: 137 –171.

Euler, L. 1744. *Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes; Additamentum I: de curvis elasticis.*

Farin, Gerald. 2002. "A History of Curves and Surfaces in CAGD." In *Handbook of Computer Aided Geometric Design*: 1–23.

Hesse, Petra, Ed. 2012. "TailorCrete, Flight Assembled Architecture." In A*rchitekturteilchen. Modulares Bauen im Digitalen Zeitalter,* 126–127, 164–165. Köln..

Jepsen, C., M. Kristensen, and P. Kirkegaard, P. 2011. "Dynamic Double Curvature Mould System." In *Computational Design Modeling: Proceedings of the Design Modeling Symposium, Berlin.* Ed. Christoph Gengnagel, Axel Kilian, Norbert Palz, and Fabian Scheurer. Springer, 291–300.

Lim, S., R.A. Buswell, T.T. Le, S.A. Austin, A.G.F. Gibb, and A. Thorpe. 2012. "Development in Construction-Scale Additive Manufacturing Processes." *Automation in Construction* 21, 1: 262–268.

Lloret Ena, Amir R. Shahab, Linus Mettler, Robert J. Flatt, Fabio Gramazio, Matthias Kohler and Silke Langenberg. 2014. "Complex Concrete Structures: Merging Existing Casting Techniques with Digital Fabrication." *Sciencedirect.com*

Pottman et al. 2007. *Architectural Geometry.* Bentley Institute Press.

Søndergaard, A., J. Feringa, T. Nørbjerg, K. Steenstrup, D. Brander, G. Gravesen, S. Markvorsen, A. Bærentzen, K. Petkov, J. Hattel, K. Clausen, K. Jensen, L. Knudsen, and J. Kortbek. 2016. "Robotic Hot-Blade Cutting." In *Robotic Fabrication in Architecture, Art and Design 2016, proceedings of RobArch 2016.*

Truesdell, C. 1983. "The Influence of Elasticity on Analysis: The Classic Heritage." *Bulletin of the American Mathematical Society* 9, 3: 293–310.

Veenendaal, D, M. West, and P. Block. 2011. "History and Overview of Fabric Formwork: Using Fabrics for Concrete Casting." S*tructural Concrete* 12, 3.

Yamaguchi, F. 1988. *Curves and Surfaces in Computer Aided Geometric Design.* Berlin Heidelberg: Springer-Verlag.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_21, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html