ITU A|Z • Vol 16 No 1 • March 2019 • 11-27

CURVED.IT: A design tool to integrate making with curved folding into digital design process

Saadet Zeynep BACINOĞLU[1], Luka PISKOREC[2] , Toni KOTNIK[3]
[1] bacinoglu@itu.edu.tr • Graduate Program of Architectural Design Computing, Institute of Science, Engineering, and Technology, Istanbul Technical University, Istanbul, Turkey
[2] luka.piskorec@aalto.fi • Department of Architecture, School of Arts, Design, and Architecture, Aalto University, Espoo, Finland
[3] toni.kotnik@aalto.fi • Department of Architecture, School of Arts, Design, and Architecture, Aalto University, Espoo, Finland

Received: July 2018 • Final Acceptance: December 2018

Abstract

The act of changing the direction of a sheet surface along a non-straight curve is a specific case of curved folding. From an architectural point of view, curved folding is an exciting operation. One or a couple of operation can generate highly complex shell-like spatial enclosure. From a digital design perspective, the implementation of curved folding with the built-in toolsets of available computer-aided design softwares is a challenging problem. The equilibrium state of curved folded geometry is needed to be found with a computational form-finding strategy. To use curved folding as a digital design operation, we introduce a new tool through developing a digital procedure for form-finding. The tool we develop can enable the experimentation with curved folding in the early stage of design process and facilitate the subsequent design development. In this article, we briefly present the literature focusing on curved folding in computational geometry, as well as the scope and description of a subclass of curved folding operation. Then, we introduce a digital tool, CURVED.IT through a design manual for its implementation and an algorithmic framework for its extension. Lastly, we discuss the design examples generated by CURVED.IT, and the potentials of the tool.

Keywords

Computational form finding, Curved folding technique, Digital design tools, Dynamic relaxation method, Surface resistant structures.

doi: 10.5505/itujfa.2019.44712

1. Introduction

This research is derived from the exciting potential of a particular technique, curved folding, for architecture and structures. A specific case of curved folding technique is the action of changing the direction of a surface along a non-straight curve on the sheet surface (Figure 1). When the sheet is folded along a non-straight curve, the three-dimensional configuration of the sheet comes to a resting state through a combination of folding and bending. While the pure folding transfigures the sheet to a polyhedral surface with an abrupt change in the surface direction, the act of bending forms the sheet to a smoothly curved surface with a gradual slight change in the surface direction. The interesting point of curved folding is the complex three-dimensional form that is generated by this hybrid surface deformation. In this form, the slender sheet gains strength and become more resistant to buckling (Figure 1).

In an architectural context, curved folding technique can be used as a design operation. The smooth and abrupt surface deformations that are generated with curved folding can provide architects with the generation of complex geometries that host different spatial situations and performative capacities. A simple example by Patkau Architects (2017) demonstrates the spatial and structural potential of the 304x731 cm folded sheet with one single fold. The folded surface is self-standing by solely touching the ground from two points (Figure 2).

In the problem of curved folding, the challenge is to understand and measure the resultant three-dimensional configuration of the curved folded sheet when it is manually modeled. The resultant three-dimensional form is "a resting state of the deformed two-dimensional sheet which goes beyond the mathematics of developable surfaces to a question of physics: equilibria of an unstrechable surface with uncreased and creased portions folding elastically toward desired angles" (Koschitz et al., 2008). Therefore, the computational form-finding strategies provide an appropriate method to find the equilibrium state of the curved folded geometry in the digital environment.

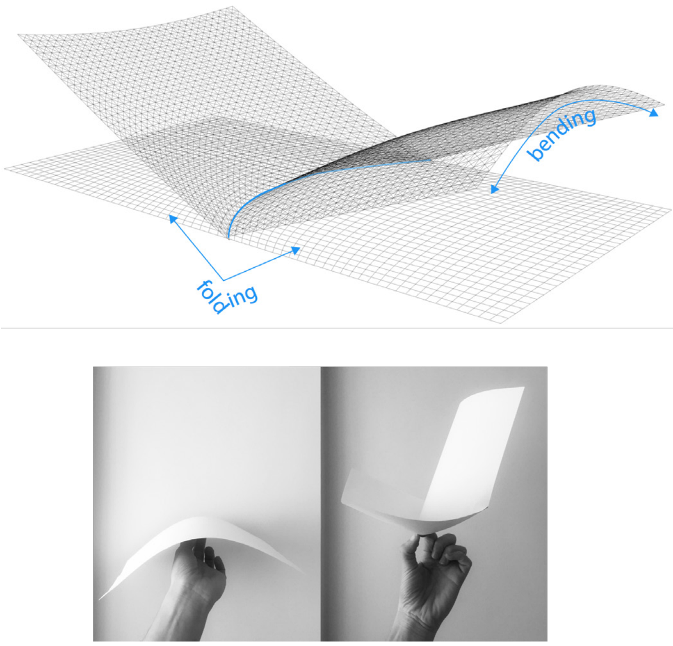

Figure 1. *Curved folded geometry.*

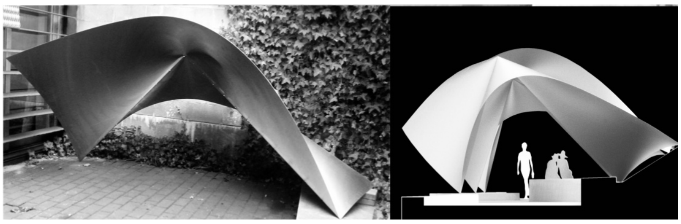

Figure 2. *Free-standing units of One Fold by Patkau Architects (2017).*

In this study, we transferred curved folding to the computer environment through developing a computational form-finding strategy in Python language within Rhinoceros environment. The main goal of this study is to introduce a design tool that allows a specific technique, curved folding, as a design operation in the computational process to explore spatial forms. The essential contribution of this study is a proposed digital form-finding tool for the architects and the designers. The proposed tool can be easily used in the early stage of the computational design process to generate curved folded geometries.

This article presents the literature focusing on curved folding in architecture and computational geometry, as well as the scope and the description of the curved folding technique. Then, the article introduces a design tool, CURVED.IT, by explaining how the proposed algorithmic framework integrates curved folding into the early stage of the computational design pro-

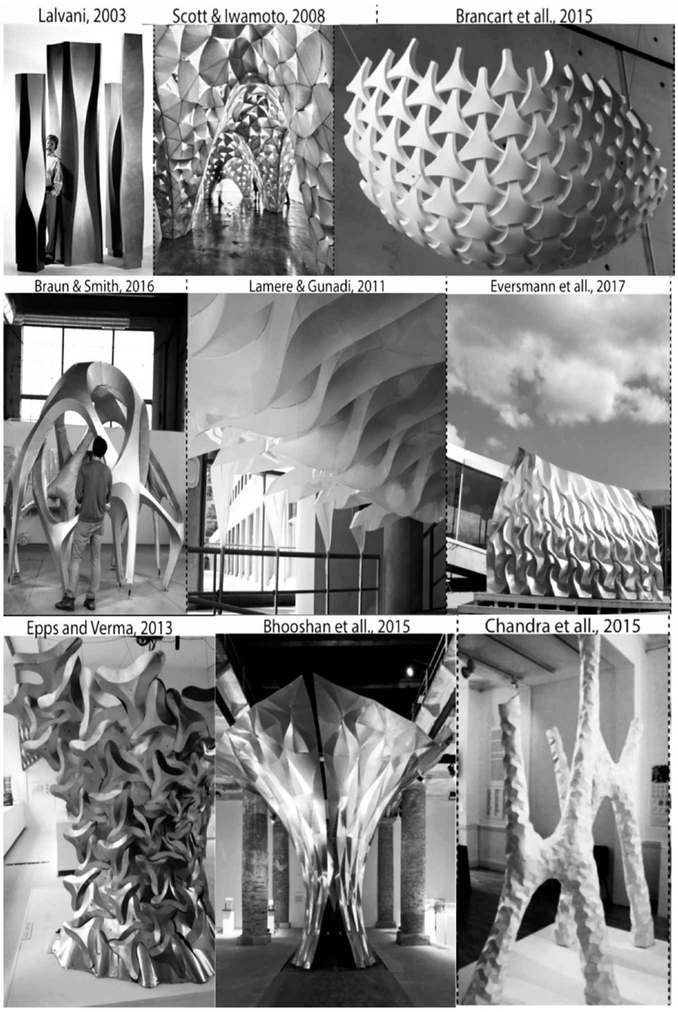

Figure 3. *Applications of curved folding in architecture.*

cess. Moreover, it presents the design manual for the architects by explaining the steps to be taken for the form generation. Lastly, it discusses the shortcomings and the potentials of the tool.

2. Studies on curved folding in architecture and mathematics

There are many applications of curved folding as computational models in the digital environment and as architectural scale installations in the physical environment. In the field of architecture, there have been increasing design researches that developed various kinds of experimentally conceptualized installations, pavilions, or building elements using the advantages of curved folding. Most of the researches have used curved folding as tessellated, small-scale surface panels. The tessellation is either applied to a predefined global geometry with a top-down manner (Lalvani, 2003; Scott and Iwamoto, 2008; Epps and Verma, 2013; Brancart et al., 2015; Chandra et al., 2015; Bhooshan et al., 2015; Eversmann et al., 2017) or the tessellated macro-architectural form is generated in an additive bottom-up process by adding a new sheet element subsequently (Lamere and Gunadi, 2011; Braun and Smith, 2016). The surface tessellations mostly function as surface strengthening, ornamenting or fabricating method. The approach of using folds on the surface of predefined forms is not distinct from previous architectural applications dating back to the 1950s. In earlier applications, many architects and engineers, such as Freyssinet, Nervi, Zehrfuss, Musmecci, and Ando, used folds mainly on the surface of predefined global forms by employing a top-down approach.

Today, the same approach has been revived with digital technologies and the availability of sheet materials, replacing the laborious concrete casting and form-work applications. The increasing availability of computer-aided design and manufacturing (CAD-CAM) reflects architecture as a phenomenon for enhancing the affordance of the surfaces through discretized surface functions such as the methods for subdividing, tessellating, or patterning. The celebrity architect, Frank Gehry, is the pioneer of this phenomenon. His office developed the software, Digital Project, to provide a set of post-rationalization tools for making the free-form geometries constructable. In the case of curved folded sheet, ARUM installation is one of the recent example (Bhooshan et al. in Figure 3, Figure 4). Before its constructions, the subdivision of the global form through curved folded pieces was calculated by Zaha Hadid's computational design research group.

Due to the difficulty of understanding the three-dimensional state of the curved folded sheet, a wide range of computational studies have developed different methods to model curved folding. We investigated the related studies across three main topics in order to gain a comprehensive understanding. Those topics are named as early mathematical descriptions, constructive geometric methods, and discrete differential geometric meth-

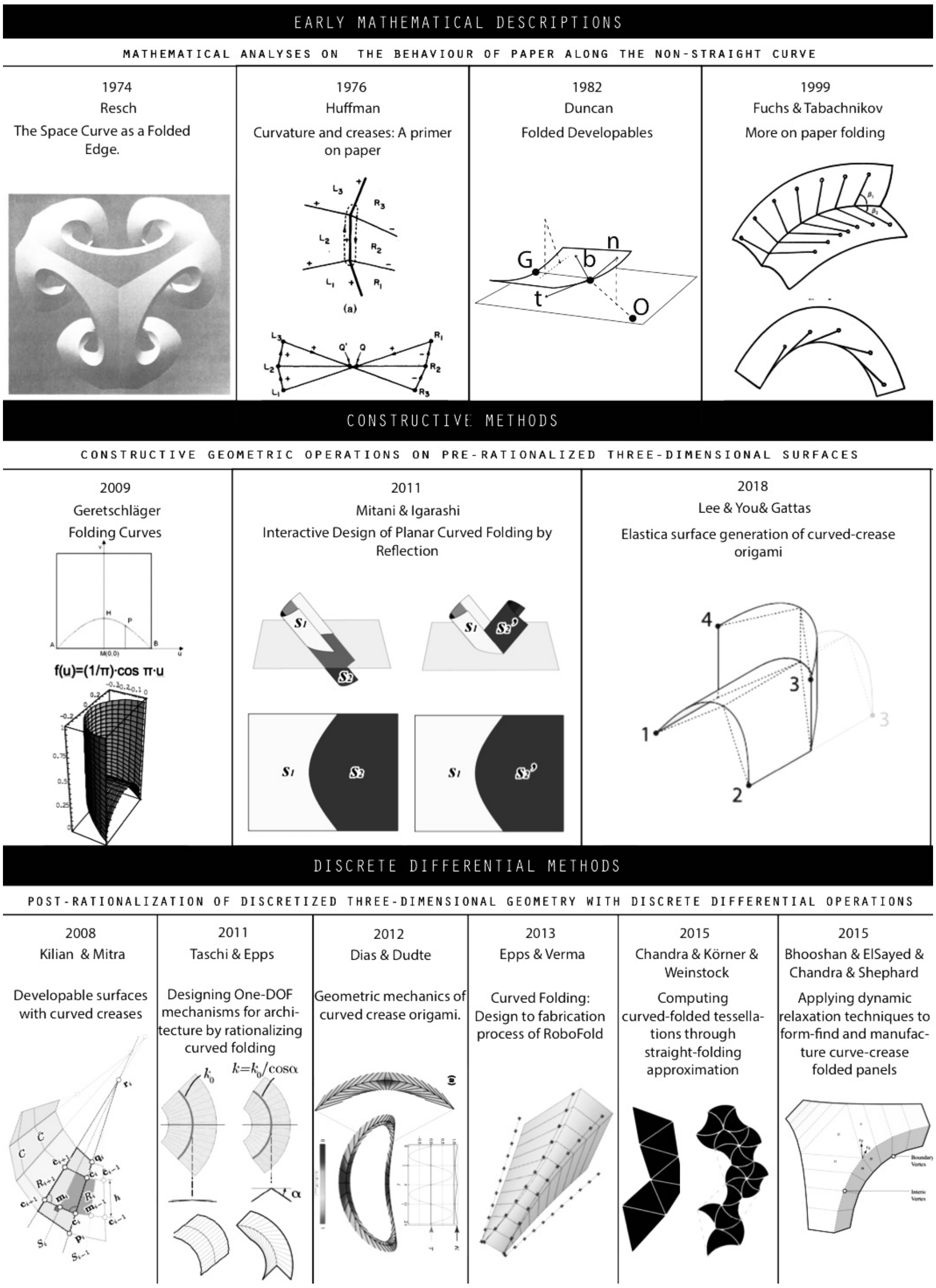

Figure 4. *Geometric approaches to represent curved folding with a computer.*

ods (Figure 4). The early mathematical studies attempted to understand and describe the behavior of the paper sheet along the non-straight curve. In mathematics, the sheet of paper is represented by a specific class of a mathematical surface, which is known as a developable surface (Pottmann, 2007). Not only the paper but also any inextensible sheet material that can elastically deform without stretching and tearing, is called a developable surface in mathematics. To describe curved folding, early studies used the characteristics of developable surfaces: the surface consists of straight lines that are either parallel to each other, intersect in a point, or tangent along a surface; the lines that generate surfaces have constant lengths; each point on the surface has a zero Gaussian curvature. These studies set mathematical equations to describe the relationship between the properties of the points on a non-straight curve, the properties of the generator lines of the paper surfaces, and the relationship between them.

Resch (1974) was one of the first computer scientists who described the curved folded surface using lines along the curve edges utilizing early computer graphics in the 1970s. In the same decade, an MIT professor, Huffman (1976), published an article on the behavior of a paper sheet near the non-straight curve. He described the interrelationship among angles and orientation of associated surfaces along a non-straight folding axis for computer-aided design and computer graphics applications. Huffman's (1976) understanding of the behavior of paper geometry was furthered by Duncan and Duncan (1982) and Fuchs and Tabachnikov (1999). The descriptions of Huffman (1976), Duncan and Duncan (1982), and Fuchs and Tabachnikov (1999) are constituted as the fundamental mathematical rules to represent the geometry of curved folded paper in computer graphics. However, Koschitz (2014) said that "there is still no real mathematical representation that can tell us where the curved creases really are in their folding states." These mathematical rules were recited by Vergauwen et al. (2014), who pointed out that "they have contributed to a better understanding of the behavior of the generator lines of the surface along a folded crease. However, they do not provide a general method to describe the folding process."

Recent works from 2008 to the present, in Figure 4, have been partly built on Hufmann, Duncan and Duncan, and Fuchs and Tabachnikov's descriptions on developable surfaces. The properties of developable surfaces have been used as geometric constraints for analytical design operations and mathematical functions in the following studies (Figure 4). The following mathematical studies mainly approached the problem of modeling curved folding by adopting two main methods: using constructive geometry or discrete differential geometry. The constructive geometric method is based on modeling curved folded geometry by applying constructive geometric transformations, such as mirror reflection and rotation operations along a curved section on the predefined developable surfaces (Mitani and Igarashi, 2011; Geretschlager, 2011; Lee et al., 2018). This method is a pre-rationalization approach to the design and modeling of curved folded geometries because it generates curved folded geometries by applying constructive geometric transformations on pre-rationalized surfaces.

The discrete differential method to attain the properties of curved folded geometry is based on the application of the mathematical functions (mostly vectors for the displacement) to the vertices or edges of a discretized (subdivided) geometry. The discrete differential approach mainly optimizes, rationalizes, or approximates the pre-defined folded state of a three-dimensional geometry with a differential operation based on geometric constraints, which mostly stem from a *priori* knowledge of developable surfaces such as "t*he sum of the angle between edges around a point must be 360 degree*" (Kilian et al., 2008; Taschi and Epps, 2011; Dias and Dudte, 2012; Epps and Verma, 2013; Chandra et al., 2015; Bhooshan et al., 2015). From a design perspective, we evaluated these two methods as pre-rationalization and post-rationalization approaches to digital modeling of curved folding. While post-rationalization approaches exclude the conceptual design phase of geometric spaces, pre-rationalization approaches include the exploratory design phase of the three-dimensional geometries. Because pre-rationalization approaches construct curved folded geometry on mathematical developable surfaces, post-rationalization approaches optimize the discretized free-form surface with developability constraints to generate curved folded geometry. Furthermore, the end user needs to have specific a *priori* knowledge to apply these methods to their design process. While post-rationalization methods require a pre-knowledge of mesh generation procedures in the digital modeling environment, pre-rationalization methods require a technical knowledge on generating appropriated topology for mathematical developable surfaces as a user input. Thus, the level of complexity of the current methods presents difficulties for user interaction.

In this brief survey on the applications of curved folding in architecture and computational geometry, the following problems were observed within the scope of this article: 1) Curved folding has been merely used as a surface enhancer in the later stage of design, rather than as a generator of the architectural space in the conceptual phase of design. 2) Curved folding has tended to be modeled precisely as a top-down method for simulation and fabrication. 3) There is a lack of CAD software to use curved folding as a design operation for architectural design exploration. 4) The proposed precise models of simulation require to have a *priori* knowledge of fabrication and material constraints, as well as mathematical developable surfaces or mesh generation procedures for user implementation. 5) The proposed computational frameworks require to have an expert technical background understanding of differential geometry for the development.

As distinct from previous architectural applications, this article achieves the following steps: 1) The study approaches curved folding as a form-finding (bottom-up) method to connect space and structural making in architecture. 2) The study does not intend to make a precise mathematical model of simulation. 3) The proposed design tool, CURVED.IT, integrates curved folding as a design operation in the early stage of computational design. 4) The tool offers a simple approach for user implementation. 5) The algorithmic framework of CURVED.IT was developed with a design research approach by translating the direct exploration of the phenomenon of curved folding in physics to digital design without requiring a *priori* specialist technical knowledge.

3. CURVED.IT: A computational design tool

3.1. Aim and scope

The main goal of developing CURVED.IT is to allow the architects and the designers the easy use and extension of curved folding in the digital medium. The proposed digital design tool, CURVED.IT, is developed through formulating an algorithmic schema of curved folding in Python language and embedding the formulated algorithm into a Rhinoceros as a tool button. The algorithm schema of CURVED.IT is developed based on preliminary observations with paper sheets which cannot stretch or shrink (Section 3.2). The inextensible characteristics of the paper, which is translated as a geometric constraint of constant local distances, form the basic idea of CURVED.IT algorithm (Section 3.3). The constraint of constant local distances on the surface is integrated into a dynamic relaxation framework (Day, 1965) using Python programming language within Rhinoceros.

3.2. The geometrical assumptions of curved folding

Before the development of CURVED.IT, the preliminary observations are made with a paper model in the physical environment and a digital model in Rhinoceros modeling environment (Figure 5). To understand the displacement of the flat sheet surface, the digital model is discretized (subdivided) into pieces of the surfaces (d). The non-straight curve on the sheet surface is used as a folding axis. The digital surface pieces (d) are folded along the non-straight curve with rotation operation in Rhinoceros.

It is observed that the pieces of surfaces (d) orient based on the direction (N) of the associated part of the curved folding axis (Figure 5-2A). In the case of straight line folding, the folding axis has a single tangent vector direction. That means that all pieces of a surface orient in one parallel direction along a straight folding axis. In the case of curved folding, the orientation of each piece along the curved axis (N) differs in a non-parallel fashion. As a result, the pieces of surfaces (d) tend to separate or converge (Figure 5-2B). The distance between pieces gets larger or shorter as distinct from an inextensible sheet deformation (Figure 5-1B). To keep the distance stable as an inextensible sheet, the opposite tension and compression forces are needed to be applied on each surface pieces. In this study, we calculated these tension and compression forces by measuring the distance between the adjacent pieces. We used these forces to guide the displacement of each interrelated piece step by step towards an equilibrium configuration of an inextensible sheet. We used the basic concept of Dynamic Relaxation to develop our

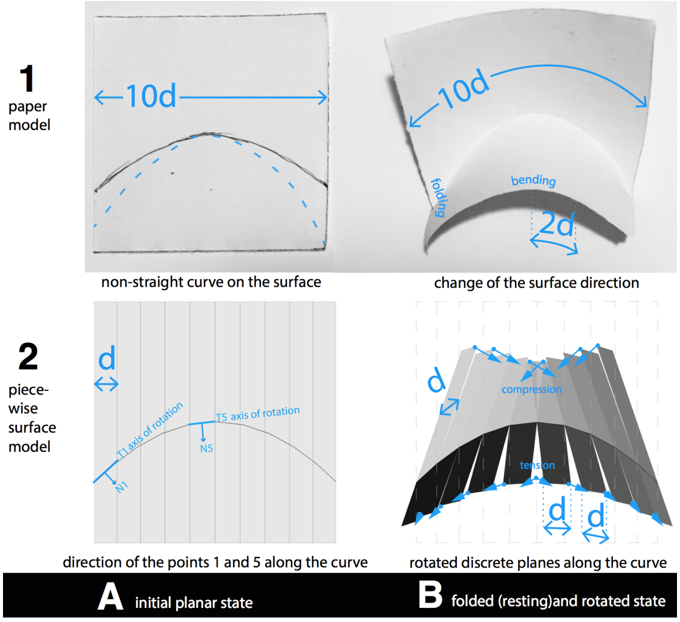

Figure 5. *Curved folding of single sheet surface and pieces of the surface.*

algorithmic framework. The numerical technique of displacing the objects towards a goal state by dividing the interval to small steps is called Dynamic Relaxation.

3.3 Algorithmic framework of CURVED.IT: A dialog with a computer

To integrate curved folding into the computational design process, we developed CURVED.IT with the algorithmic framework of dynamic relaxation which is built on the preservation of the distances between pieces after they are folded along the non-straight curve. For a more exact measurement, we defined the inextensible sheet material is as a point network. The point network is abstracted as a particle-spring system.

Particle-spring system is a discrete differential procedures. In this system, particles are point objects that have properties of mass, position, and velocity. Particles can be made to exhibit a wide range of interesting behaviors. Springs are the connections between particles. As vector objects, springs basically define the behaviors of the particles. In our case, we programmed spring vectors to generate additional vectoral forces to move each particle towards an equilibrium position. The particles arrive an equilibrium position when the distance between each point in the deformed state became same as the distance value in their flat state. As we mentioned in the previous section 3.2, the distance between each neighbor particle shrinks or stretches when the particle-spring system is deformed along a non-straight curve with an applied rotation operation. With spring vector function, we measure the residual force (length defect=D-d) at each vertex. Subsequently, we multiply the residual force with the damping factor to displace each particle (deformed point) incrementally towards the initial length values (Figure 6). We repeat the previous operation until the distance between each connected point arrives at the initial length (d) where the point network finds the equilibrium solution. This step-by-step small displacement of points with an iterative calculation to find the equilibrium position of each point in the network is called as Dynamic Relaxation (Day, 1965). This numerical form finding technique has been widely used by structural engineers to find the equilibrium state of structures (Sutherland, 1963; Barnes, 1977; Williams, 2001; Kilian and Ochsendorf, 2005)

We developed CURVED.IT using the Python object-oriented programming language within a Rhinoceros three-dimensional modeling environment. The algorithmic process of CURVED.IT is described schematically in Figure 7. The algorithm includes six main steps. The steps are as follows:

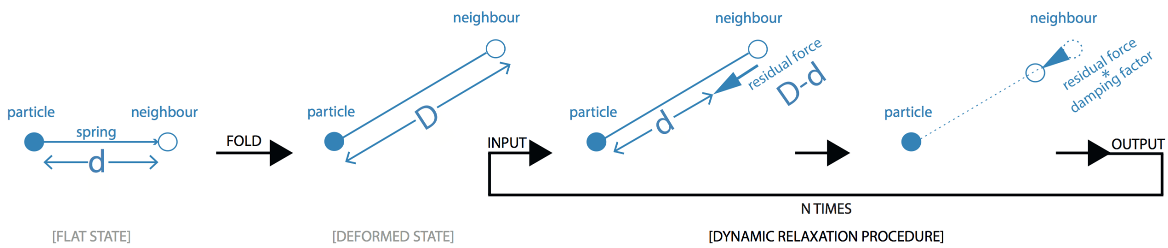

Figure 6. *The diagram of the behavior of one particle with its one neighbor in the CURVED.IT algorithm.*

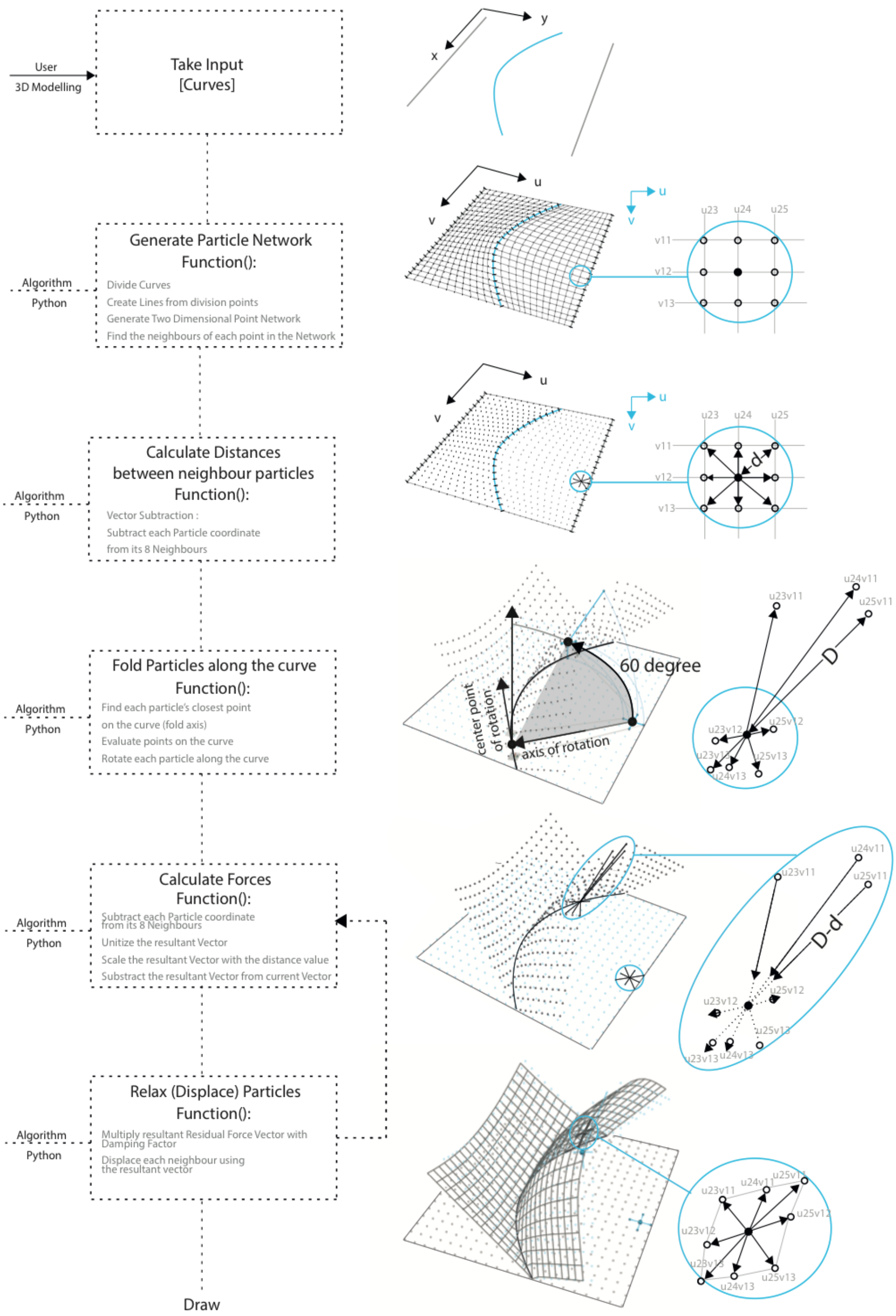

Figure 7. Steps of the dynamic relaxation (DR) algorithm.

3.3.1. Obtain objects (Curves)

First, the two-dimensional curves are input from Rhinoceros modeling environment into Python.

3.3.2. Generate a particle network

Secondly, the two-dimensional curves that the user input are subdivided with a nested function and subsequently a two-dimensional point array (matrix) which consists of rows (v) and columns (u) is generated. With this two-dimensional hierarchy, the orthogonal and diagonal neighbors of each point become accessible by checking each point's position in rows (v) and columns (u).

3.3.3. Calculate the distances (d)

Thirdly, the distance between each point and its orthogonal and diagonal neighbors (d) in this array is measured by subtracting the coordinates of the point from each of its neighbors. The distances in the flat state of the sheet (d) are stored in a matrix table.

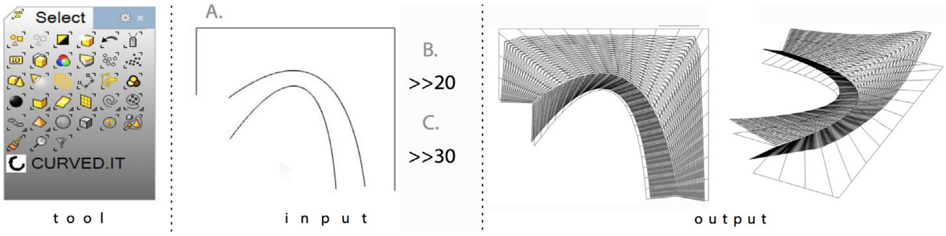

Figure 8. The access to CURVE.IT in Rhinoceros environment.

3.3.4. Fold the particle network (Rotate)

Fourthly, the point array along the non-straight curve is rotated with Fold function. To create Fold, the closest point of each point on a curved axis and the tangent vector are defined as the rotation center point and rotation axis to rotate each point a variable (60) degree along the curve.

3.3.5. Calculate the residual force (D-d) with an iterative measuring process

Later, the distance between each point and its neighbors is calculated (D). To keep the distance constant, the difference between the deformed state and the initial state (D-d) is calculated using the vector subtraction operation. The difference between the length between the two points in the initial state and the length in the deformed state gives the residual force vectors.

3.3.6. Relax with an iterative displacement process

Lastly, the residual forces (D-d) guide the relaxation process. The residual force is decreased by multiplying it with a damping factor to create small changes in the displacement between successive iterations (to prevent big jumps). In each iteration, the algorithm recalculates the distance between points and their neighbors, and the resulting residual force vector displaces the points towards the expected solution. The expected solution is to arrive at the initial distance value between points. The iteration continues until the solution reaches an equilibrium state, where the net residual force on each node equal to zero.

3.4. The steps to be taken with CURVED.IT: The design manual

Rhinoceros software allows its user to extend the software through user-defined specific procedures. These custom procedures can be brought to the software's toolbar as a button. We embed the developed algorithm that is described in the previous section 3.3 in Rhinoceros toolbar (Figure 8). After the user installs CURVED.IT code, he/she can access the tool with one click to the customized button in Rhinoceros toolbar.

The customized tool, CURVED.IT, allows the use of the specific case of curved folding as a design operation in the early stage of the computational design process. The design tool virtually folds the user-defined surface along the user-defined non-straight curve. By clicking CURVED.IT button, the user can follow the instructions from the command line. The user of the tool can easily create a curved folded surface following three main instructions (Figure 9). The instructions are as follows: 1) Determine the boundary edge curves and the fold curve (The user draws two-dimensional edge curve (curve1), fold curve (curve2), edge curve (curve3) in a sequence by using the three-dimensional modeling software Rhinoceros's in-built curve commands. When the user hit enter, the curves input to Python module. Then the user reselects the fold curve and press enter to input the curves to Python module). 2) Determine the density of the point network (The user inputs a number value to the command line as a curve division parameter). 3) Determine the fold angle (The user inputs a number value to the command line for the fold angle).

After inputting data by selection and insertion into the command line, the folded geometry is calculated and drawn as curves between points in 3d Rhinoceros environment.

The simplicity of the tool allows the user to explore diverse spatial configurations by playing with two or more

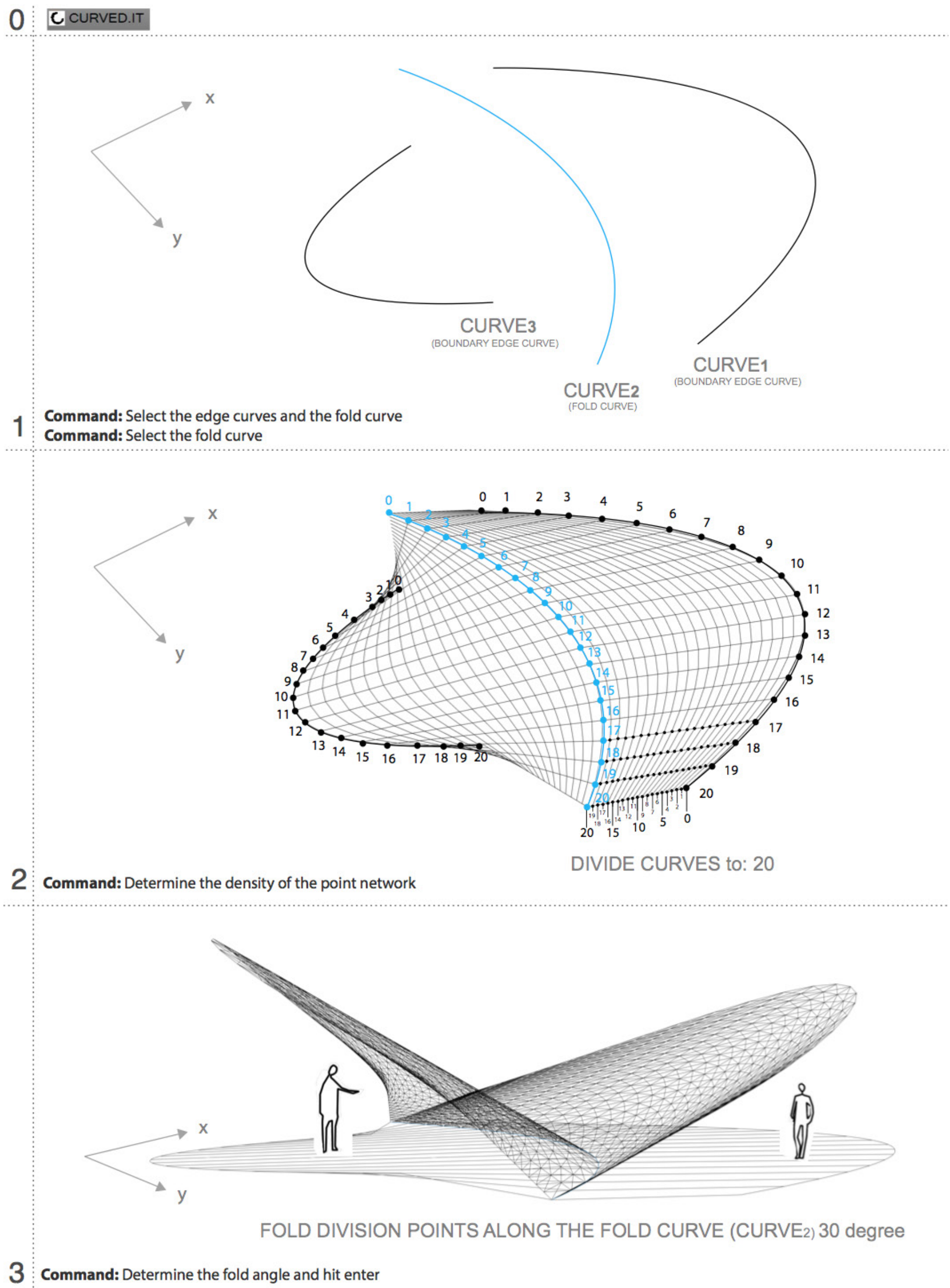

Figure 9. *Generation of a curved folded space with three instructions of CURVED.IT.*

curves on the flat surface. The user can adjust the properties of the free-form curve, the relationships between the curves, the density of the point network, and the degree of the fold angle to generate three-dimensional curved spaces. The properties such as curvature of a curve, the length of a curve, and the shape of a curve can be altered by changing the position and the number of control points of the NURBS curve or changing the degree of curve. The in-built command of Rhinoceros turns on the control points and the curvature graph of the curve to edit control points and the degree number. The simple variations of input curves and the resultant form can be seen in Figure 10.

In this process, CURVED.IT aids the user by quickly calculating the three-dimensional folded configuration of the user determined curves. Moreover, the resultant form can be considered as a feedback to a designer. The designer can further the design process or repeat the same operation with a different input configuration. The designer can use CURVED.IT iteratively as a design operation through an ongoing process. To continue the iterative process with CURVED.IT.,

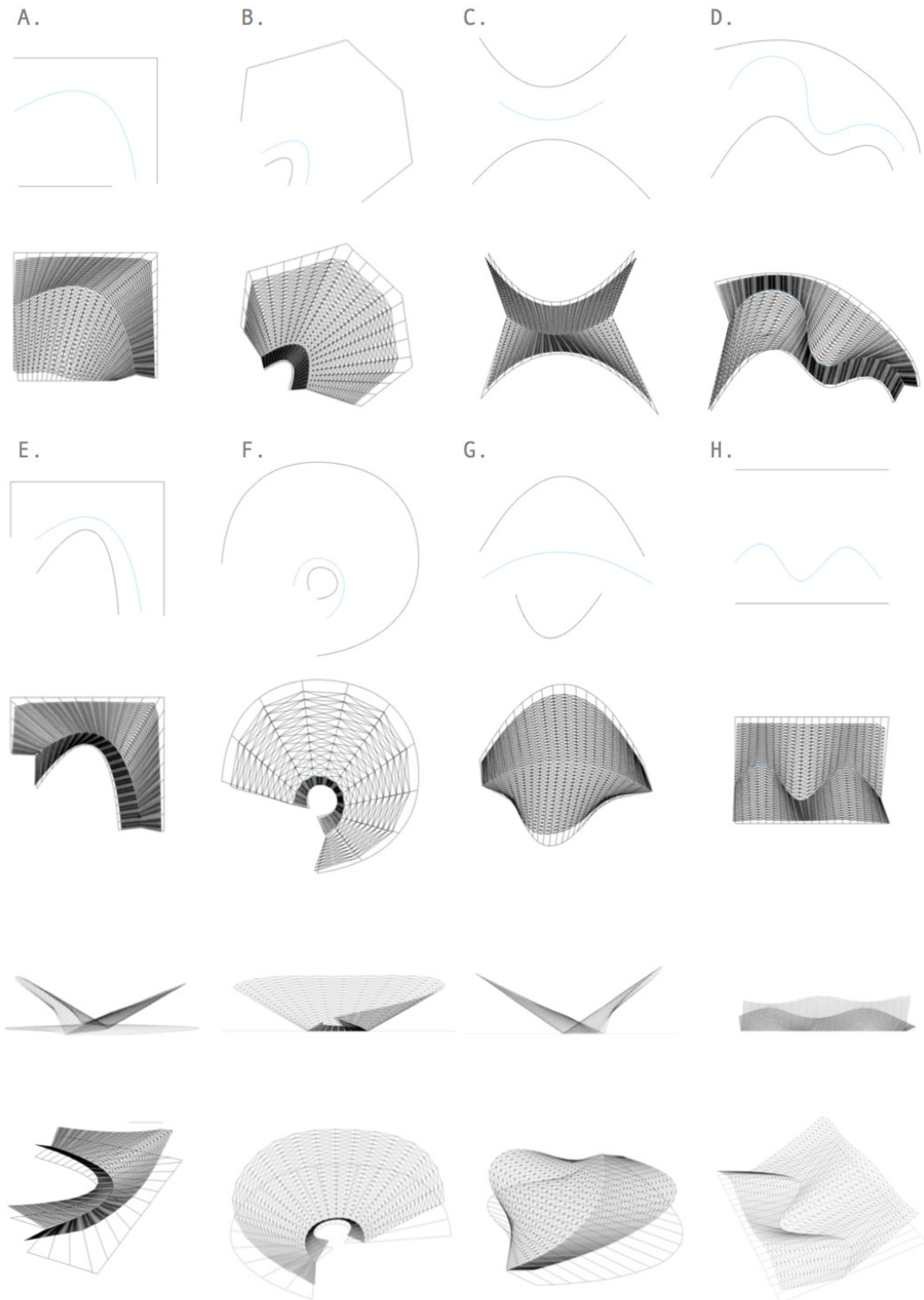

Figure 10. *A simple demonstration of variations by changing the combination of three simple curves.*

the user selects one folded edge of the curved form (c3') as a new fold curve and draw a new curve (c4) as a new edge of the surface. After input the new curves to CURVED.IT., the new folded surface is added to the previous form. If the user does not be satisfied with the 3-D resultant form that is generated by CURVED.IT, he/she can step back and edit the curve (c4= boundary curve) and fold it again as in Figure 11. The user can complete the process when she arrives a satisfied state.

4. Potentials of the tool

The proposed design tool, CURVED.IT, integrates curved folding to the Rhinoceros three-dimensional modeling environment as a form-giving design operation. The simplicity of the tool can provide the novice designer an easy acquaintance with the digital design process, as well as with the technique of curved folding. The user can explore the diverse spatial configurations by playing with the two-dimensional curves. The resultant spatial configuration that is generated with CURVED.

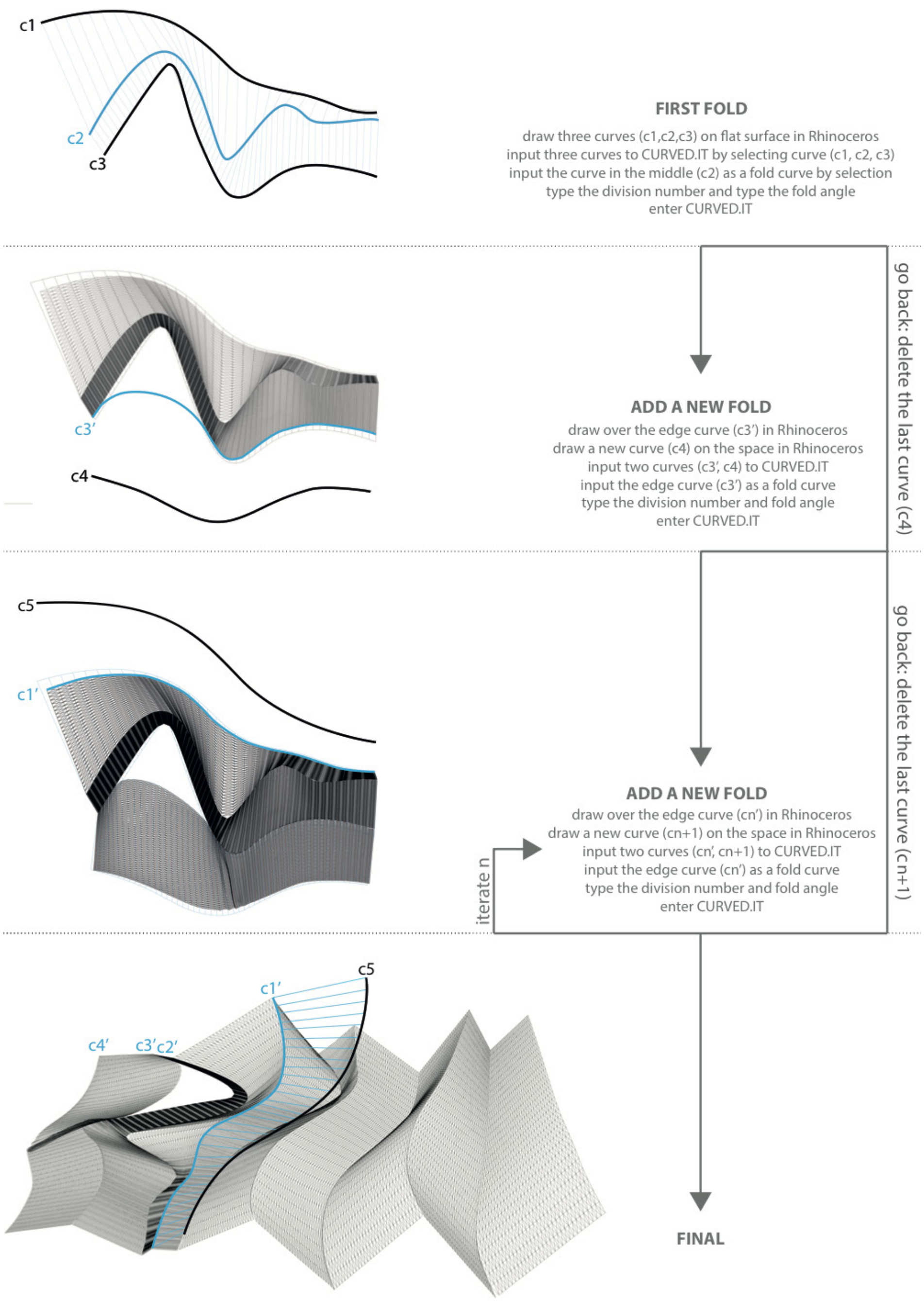

Figure 11. *Demonstration of the addition of new folds step by step.*

IT is consists of lines between points. The abstract configuration of lines can be furthered by the designer according to his/her design intentions. Such as the lines can be transformed to pipes or the surface panels can be created in between the lines. Figure 12 demonstrates the furthered geometry (E) which was shown in Figure 10. Similarly, Figure 13 shows the interpretation of geometry (H) as an architectural shell structure. One of the advantages of generating the form that is consist of multiple standard elements is the ease of application of the same operation to many elements with one click. As seen in Figure 14, the numerous joint geometry was produced with one operation.

Currently, CURVED.IT allows the easy use of parallel curved folds as seen in previous figures. However, the algorithm has not extended for the simultaneous folding of multiple non-parallel curved pieces. An example in Figure 15 shows a spatial configuration of four curved folded space and the connection in between. In this example, we input the two-dimensional curves

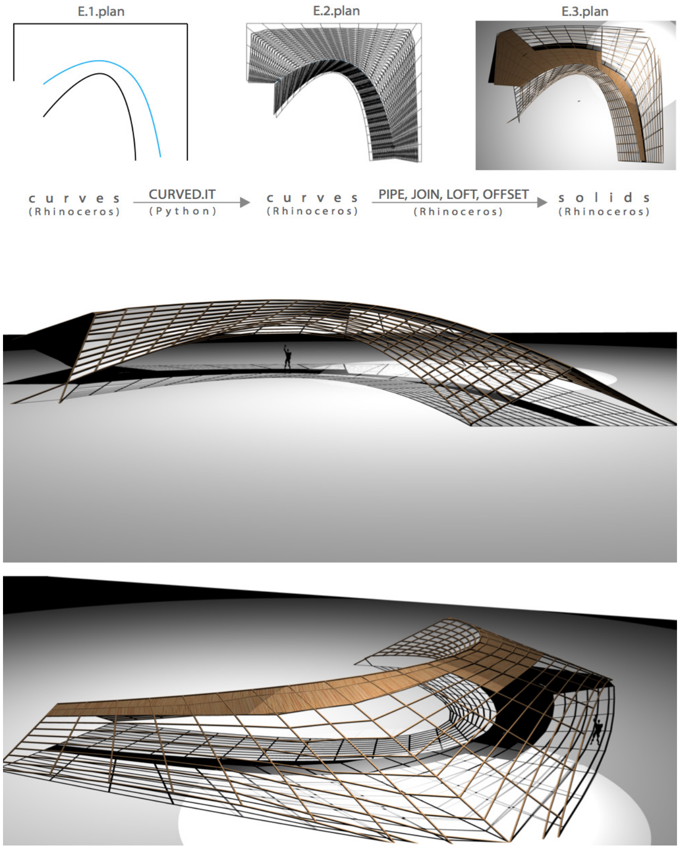

Figure 12. *The applied pipe and surface functions in Rhinoceros to CURVED.IT form.*

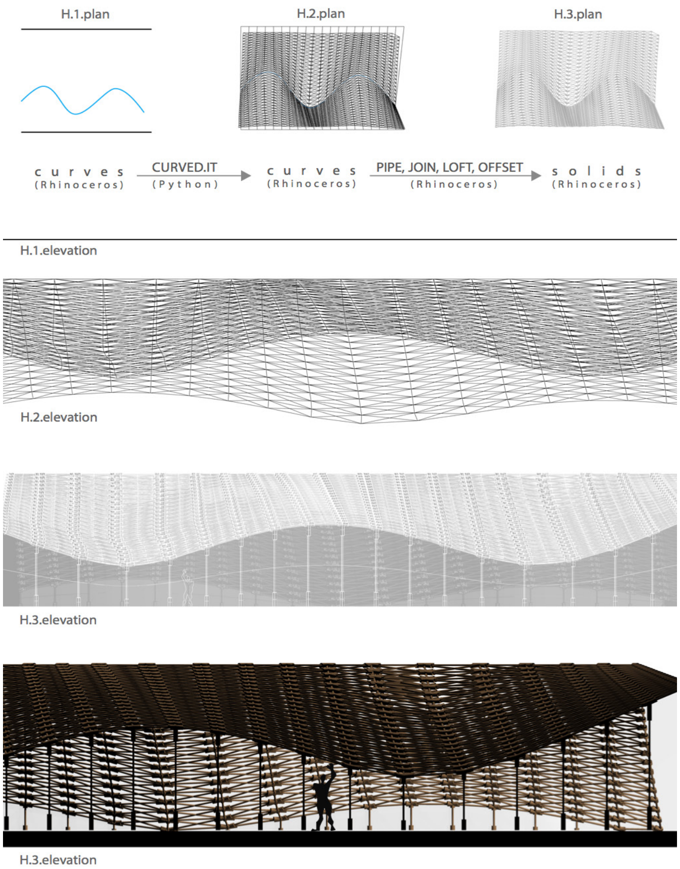

Figure 13. *The process of transforming the resultant set of lines to the solid geometries.to CURVED.IT form.*

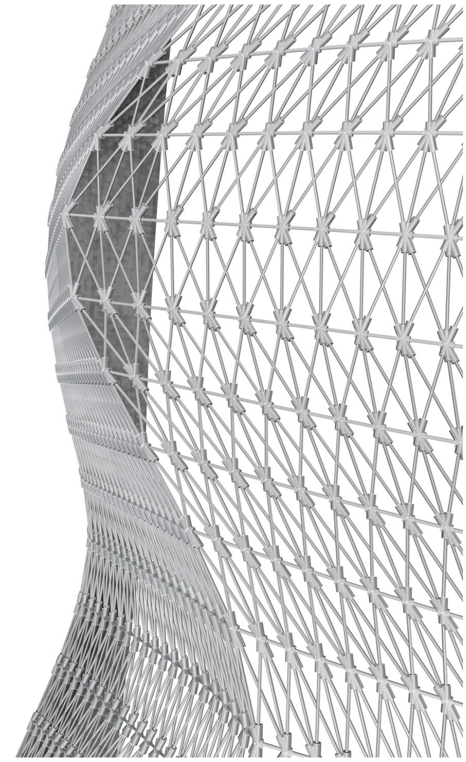

Figure 14. *The generation of a joint detail by applying the in-built Rhinoceros pipe function to the one-tenth of each line segment.*

(c1.f, c1.b; c2.f, c2.b, c3.f, c3.b, c4.f, c4.b) to CURVED.IT. The tool found the equilibrium state of curved folded pieces separately. Then, we connected the folded pieced in Rhinoceros. In the future, the algorithm can be extended through grouping the point array for the calculation of multiple folds simultaneously.

In the algorithm of CURVED.IT, the inextensibility of the sheet is abstracted as a geometric constraint which guides the interaction between particles and eventually, it relaxes the particle system as curved folded form. In this case, the tool is limited to the generation of curved folded surfaces. However, an expert user with programming skills can change the algorithm schema to generate different behaviors by altering the interactions between particles. For example, programming the net spring force target (vector sum) on each particle as a zero value can produce minimal surfaces, which initially gained the attention of architects courtesy of Frei Otto's soap film experiments.

Thus, the same algorithm can be edited to find diverse topologies which have different levels of structural com-

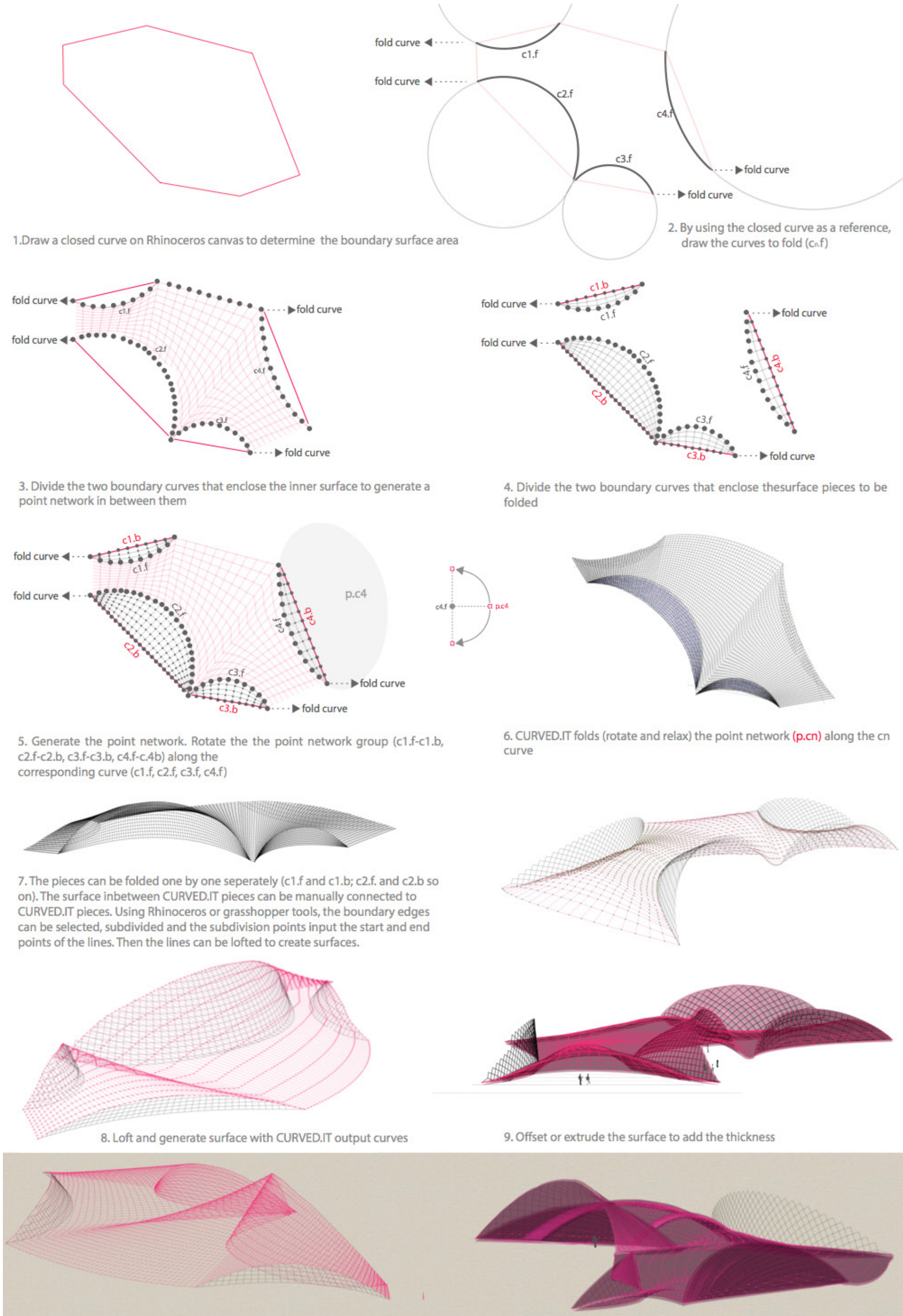

Figure 15. *The experiment with multiple non-parallel curved folds.*

plexities. The specificity and technicality of programming-based design require an advanced logical and mathematical a priori knowledge. However, such digital tools, which attract the attention of the user, potentially improve his/her knowledge. The user, who is engaged in a tool and its open code, can explore the underlying structure and rules. With the integration of the designer's conceptual design thinking, the algorithm can be extended or reformulated to capture and generate a different phenomenon. As Burry (2011) points out, many architects who use digital tools are becoming tool makers today.

5. Discussion of future applications

Today, one of the contemporary problems with digital technologies in the architectural field is the post-rationalization processes of the free-form architectures for the structural and constructional requirements (Scheurer, 2005; Pottmann et al., 2007). One of the solutions to this problem as Bollinger and Grohman (2004) points out that "to shift architectural design from pure modeling to the understanding of organizational principles and systems with a specific behavior. Solutions derived from this process do not necessarily match conventional structural systems, but they gain performance by self-organization of its members". The proposed design tool in this article is based on an organizational principle which self-organize its members to generate curved folded geometries. Since this organizational principle is defined by a computational logic, it has the potential to allow for a better link between conceptual, structural, and constructional levels of architecture.

The use of curved folding itself in the computational design process has already overlay conceptual and structural levels of architecture. The resultant bended and folded surface not only embraces a space after it is folded, but also it transforms the slender sheets into more resistant structures concurrently. Namely, it creates a shell structure. A shell is, according to Williams (2014), "a rigid structure defined by a curved surface. It is thin in the direction perpendicular to the surface. The minimal cross-section of a shell allows for material efficiency. " In summary, the action of curved folding overlays space and structure by generating the geometry of shell using minimal cross-section. In the architectural design process, curved folding presents us a geometric design method for finding lightweight architectural shell geometries. American Concrete Institute (2008) defined common types of thin shells as domes, cylindrical shells, conoids, elliptical paraboloids, hyperbolic paraboloids, and groin vaults. However, the operation of curved folding allows us to go beyond specific archetypes to complex irregular formations. Thus, the deformation generated by the operation of curved folding can actively be used in the architectural design process to explore complex spatial structures.

In this study, we developed CURVED.IT as a tool for design idea generation. Neither CURVED.IT is a tool for simulation nor the digitally-found form created with CURVED.IT is a precise model for the construction. In precise models, the accessive amount of information can slow down the early exploration processes. However, the existence of a computational model can ease the link between conception and later phases such as construction. Due to the explicit structure of the computational design, the conceptual model can be easily connected to other specialist knowledge later. For architectural scale applications, the proposed tool can be fed material information by extending the current algorithm of CURVED.IT with the code blocks on geometrical constraints between the neighbor nodes, as well as integrating the model with the available commercial finite element modeling (FEM) simulation software.

Acknowledgments

This study is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under the 2214/A- International Research Fellowship Programme.

References

ACI Committee, American Concrete Institute, & International Organization for Standardization. (2008). Building code requirements for structural concrete (ACI 318-08) and commentary. American Concrete Institute.

Architects, P. (2017). Patkau Architects: Material Operations. Princeton Architectural Press.

Barnes, M. R. (1977). Form finding and analysis of tension space structures by dynamic relaxation (Doctoral dissertation, City University London).

Brancart, S., Vergauwen, A., Roovers, K., Van Den Bremt, D., De Laet, L., & De Temmerman, N. (2015). UNDULATUS: design and fabrication of a self-interlocking modular shell structure based on curved-line folding. In Future visions; Proc. intern. symp., Amsterdam, 17-20 August 2015.

Bhooshan, S., Bhooshan, V., El-Sayed, M., Chandra, S., Richens, P., & Shepherd, P. (2015). Applying dynamic relaxation techniques to form-find and manufacture curve-crease folded panels. *Simulation*, *91*(9), 773-786.

Bollinger, K., Bollinger, K., Grohmann, M., & Schmal, P. C. (2004). Workflow: Struktur-Architektur. Brikhäuser.

Burry, M. (2011). Scripting cultures: Architectural design and programming. John Wiley & Sons.

Chandra, S., Körner, A., Koronaki, A., Spiteri, R., Amin, R., Kowli, S., & Weinstock, M. (2015, April). Computing curved-folded tessellations through straight-folding approximation. In *Proceedings of the Symposium on Simulation for Architecture & Urban Design* (pp. 152-159). Society for Computer Simulation International.

Day, A. S. (1965). An introduction to dynamic relaxation(Dynamic relaxation method for structural analysis, using computer to calculate internal forces following development from initially unloaded state). *the engineer*, *219*, 218-221.

Dias, M. A., Dudte, L. H., Mahadevan, L., & Santangelo, C. D. (2012). Geometric mechanics of curved crease origami. *Physical review letters*, *109*(11), 114301.

Duncan, J. P., & Duncan, J. L. (1982). Folded developables. Proc. R. Soc. Lond. A, 383(1784), 191-205.

Epps, G., & Verma, S. (2013). Curved Folding: Design to fabrication process of RoboFold. *Shape Modeling International 2013*, 75.

Eversmann, P., Ehret, P., & Ihde, A. (2017). 'Curved-folding of thin aluminium plates: towards structural multipanel shells'. *Proceedings of the International Association for Shell and Spatial Structures*.

Fuchs, D., & Tabachnikov, S. (1999). More on paper folding. *The American Mathematical Monthly*, *106*(1), 27-35.

Geretschläger, R. (2009). Folding Curves. *In Origami 4, Lang, R. J. (Ed.). CRC Press,* 151.

Huffman, D. A. (1976). Curvature and creases: A primer on paper. IEEE Transactions on computers, (10), 1010-1019.

Kilian, M., Flöry, S., Chen, Z., Mitra, N. J., Sheffer, A., & Pottmann, H. (2008). Curved folding. *ACM transactions on graphics (TOG)*, *27*(3), 75.

Kilian, A., & Ochsendorf, J. (2005). Particle-spring systems for structural form finding. *Journal of the international association for shell and spatial structures*, 46(2), 77-84.

Koschitz, D., Demaine, E. D., & Demaine, M. L. (2008). Curved Crease Origami, *Proceedings of the Advances in Architectural Geometry, Vienna, Austria, Sept,* 29-32.

Lalvani Haresh, 2003.

URL: http://www.metropolismag.com/uncategorized/bend-the-rules-of-structure/

Lamere, J., Gunadi, C. (2011). Overliner.

URL: http://www.gldarch.com/projects/show?utf8=✓&tag=16&project=3

Lee, T. U., You, Z., & Gattas, J. M. (2018). Elastica surface generation of curved-crease origami. *International Journal of Solids and Structures*.

Mitani, J and Igarashi T. (2011). Interactive Design of Planar Curved Folding by Reflection. In: *the 19th Pacific conference on computer graphics and applications*. Kaohsiung, Taiwan: Pacific Graphics.

Pottmann, H. (2007). Architectural geometry (Vol. 10). Bentley Institute Press.

Resch, R. D. (1974). The Space Curve as a Folded Edge. In Computer Aided Geometric Design (pp. 255-258).

Scheurer, F., Schindler, C., & Braach, M. (2005). From design to production: Three complex structures materialised in wood. In *6th International Conference Generative Art*.

Scott, C., & Iwamoto, L. (2012). Voussoir Cloud. In *Matter: Material Processes in Architectural Production* (pp. 68-80). Routledge.

Sutherland, I. (1963). SKETCHPAD-a man-machine graphical interface (Doctoral dissertation, PhD thesis, MIT).

Tachi, T., & Epps, G. (2011, March). Designing One-DOF mechanisms for architecture by rationalizing curved folding. In *International Symposium on Algorithmic Design for Architecture and Urban Design (ALGODE-AIJ). Tokyo*.

Vergauwen, A., De Temmerman, N., & De Laet, L. (2014). Digital modelling of deployable structures based on curved-line folding. In *Proceedings oftheIASS-SLTE2014Symposium "Shells, Membranes and Spatial Structures: Footprints*.

Vergauwen, A., De Laet, L., & De Temmerman, N. (2017). Computational modelling methods for pliable structures based on curved-line folding. *Computer-Aided Design*, *83*, 51-63.

Williams, C. J. (2001). The analytic and numerical definition of the geometry of the British Museum Great Court Roof.

Williams, C. (2014). What is a shell. Shell structures for architecture: form finding and optimization, 21-31.