

Iterative Geometric Design for Architecture

THÈSE N° 4572 (2009)

PRÉSENTÉE LE 13 JANVIER 2010

À LA FACULTÉ ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT
LABORATOIRE DE CONSTRUCTION EN BOIS
PROGRAMME DOCTORAL EN STRUCTURES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Ivo STOTZ

acceptée sur proposition du jury:

Prof. B. Marchand, président du jury
Prof. Y. Weinand, directeur de thèse
Prof. D. Dietz, rapporteur
Prof. R. Motro, rapporteur
Dr E. Tosan, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2009

Abstract

This work investigates on computer aided integrated architectural design and production. The aim is to provide integral solutions for the design and the production of geometrically complex free-form architecture. Investigations on computer aided geometric design and integrated manufacturing are carried out with equal importance. This research is considering an integral and interdisciplinary approach, including computer science, mathematics and architecture.

Inspired by fractal geometry, the IFS formalism is studied with regards to discrete architectural geometric design. The geometric design method studied provides new shape control possibilities unifying two separate design paradigms of rough and smooth objects. Capable to design fractal geometric figures, the method also covers the generation of classical objects such as conics and NURBS-curves. Close attention has been paid to the design of iterative free-form surfaces, which are composed entirely out of planar elements. A surface method based on projected vector sums is proposed. The resulting geometric figures are expressed in a discrete form and can be easily translated into a coherent set of constructional elements.

The studies for translation of the geometrical elements into constructional elements consider integrated manufacturing. Addressing and numbering of the elements by iterative geometric design are investigated and compared to lexicographically ordered addressing systems, in order to provide an adequate data structure for the design, production and assembly of the constructional elements. For the generation of the data describing constructional elements, problems related to thickening and offset meshes are discussed. Once the global geometry of the constructional part has been computed, parameters are defined for generic automated detailing. Hereby the entire description of the constructional elements is completed. These elements are mapped and packed with regards to the coordinate system of a CNC-machine and the properties and the dimensions of the raw material, providing the complete set of workshop plans needed for integrated manufacturing. For automated generation of machine instructions (G-code), machining strategies - depending on the type of machine used, tool and material properties - are elaborated.

Finally, the integrated digital design methods studied within the scope of this thesis are tested and verified by the realization of different reduced scale prototypes. The studied applications range from bearing vault structures to fractal and smooth timber panel shell structures. The developed methods have shown to be efficient for the design and the realization

of geometrically complex architectural objects. The required planning effort to handle and manipulate the design and the production data has been greatly reduced. Some of the proposed methods have proved to be robust and general enough to be applied on real world applications. Iterative geometric design provides high degree of design possibilities offering an efficient tool for the creation of smooth and rough free form objects. The possibility to incorporate successive folds in free-form objects allows structural applications.

Keywords: Free-form architecture, discrete computer aided geometric design, IFS, integrated manufacturing

Résumé

Ce travail s'intéresse à la question de la conception et de la production de projets architecturaux assistés par ordinateur. Le but est de développer des solutions techniques pour le dessin et la production de projet architecturaux complexes se basant sur des géométries non-standards. La recherche se concentre à la fois sur des problèmes liés à la modélisation géométrique assistée par ordinateur et la production automatisée des éléments de constructions. Cette recherche considère une approche intégrale – du dessin à l'objet – et interdisciplinaire intégrant les domaines d'informatique, mathématique et architecture.

Inspiré par la géométrie fractale, des systèmes de fonctions itérés (IFS) sont étudiés comme formalisme de modélisation géométrique discrète. La méthode de modélisation géométrique étudiée offre de nouvelles possibilités de contrôle de la forme en unifiant le domaine des formes lisses avec le domaine de formes rugueuses. Capable de concevoir des objets fractals, la méthode permet également la génération d'objets lisses comme par exemple des NURBS ou des coniques. Des études particulières portent sur la modélisation itérative de surfaces libres composées uniquement d'éléments plans. Ainsi, une méthode de modélisation de surface est développée, basée sur la somme vectorielle projetée de deux courbes. Les objets géométriques qui en résultent sont exprimés sous une forme discrète et peuvent ainsi facilement être traduits dans un ensemble cohérent d'éléments de construction.

Les études permettant la traduction des éléments géométriques en éléments de construction considèrent des procédés de fabrication automatique assistée par ordinateur. L'adressage des éléments obtenus par l'algorithme de construction géométrique itérative est étudié et comparé à des systèmes d'adressage ordonnés de manière lexicographique, afin d'obtenir

une structure des données adéquate à la conception, la production et au montage. En ce qui concerne la description géométrique des éléments de construction, le problème d'épaississement de maillage à base de carreau est abordé. Après la génération de la géométrie globale des éléments de construction, un ensemble de paramètres permet la génération automatique de détails génériques. Ainsi, les données complètes des éléments de construction sont établies. Par la suite, ces données sont transposées de l'espace de dessin 3D vers l'espace de travail de la machine à commande numérique en fonction des dimensions et des propriétés du matériau brut. A cet état, les données représentent le jeu complet des plans d'atelier pour une fabrication assistée par ordinateur. Pour la génération automatique des instructions machine (code-G), une série de stratégies d'usinage est élaborée prenant en compte le type de machine, les outils utilisés, le matériau choisi et de la géométrie des pièces.

Les outils de conception digitale développés lors de cette recherche sont testés et vérifiés par la réalisation de plusieurs prototypes. Les applications proposées varient entre des éléments ornementaux et décoratifs et des applications structurelles de coque plissée en panneaux de bois massifs. Ainsi, les outils développés doivent démontrer leur efficacité pour la conception et la production d'objets architecturaux géométriquement complexes. Les résultats montrent que l'effort de planification nécessaire pour la gestion et la manipulation des données de projet et de production a été fortement réduit. Quelques méthodes proposées ont même prouvé être assez générales et solides pour être appliquées à des projets réels. La modélisation géométrique itérative offre des grandes libertés de conception de formes donnant un outil efficace pour la création et la réalisation d'objets non standard lisses ou rugueux. Par la suite, la possibilité de pouvoir intégrer des plis successifs aux formes libres permet en plus des applications structurelles.

Mots clés : Architecture non-standard, modélisation géométrique discrète assistée par ordinateur, IFS, fabrication par commande numérique

Acknowledgments

I honestly would like to thank all the persons having contributed in the one or the other way to make this work exist in its present form:

Professor Yves Weinand, IBOIS - EPFL
Professor Bruno Marchand, LTH2-SAR - EPFL
Professor Dieter Dietz, ALICE, ENAC - EPFL
Professor René Motro, IASS, LMGC - Montpellier
Professor Peter Buser, IGAT - EPFL
Dr. HDR Eric Tosan, LIRIS - Université Lyon
Dr. es. sciences Bernhard Stamm
Dipl. Ing. DEA Gilles Gouaty, IBOIS - EPFL
Dipl. Ing. DEA Iver Bailly-Salins, Daussault Systems
Dipl. Ing. Johannes Natterer, IBOIS - EPFL
Alice Vevesny, ETU - SAR - EPFL
Benoît Cochaud, ETU - INSA - Toulouse
Milo and Sara Gerber, my Family

Further, I would like to thank the Swiss National Foundation for supporting financially the project “Fractal Geometry and it’s application in the field of timber construction” (FNS 200021-112103 and 200020-120037/1).

Contents

Abstract	3
Résumé	4
Acknowledgments	6
Part 1. Introduction	13
Chapter 1. Goals and Methods	15
1.1. Geometric Design	15
1.1.1. Iterative Geometric Design	15
1.1.2. Control Parameters	16
1.1.3. Constraints	16
1.2. Digital Production of Architecture	17
1.3. Applications of Discrete Geometry to Architecture	18
Chapter 2. State of the Art	19
2.1. Architectural Geometry	19
2.1.1. On Architectural Geometry	19
2.1.2. Mathematical Models for Architectural Use	20
2.2. New Methods in Timber Construction	22
2.2.1. Integrated Manufacturing	22
2.2.2. New Timber Products	23
2.2.3. New Timber Structures	23
2.3. Broader Framework of the Research	24
2.3.1. Interdisciplinary Research Team	24
2.4. Existing Work in Fractal Geometry	25
2.4.1. Applications of Fractal Geometry	25
2.4.2. Fractals in Architecture	25
2.4.3. Fractal Geometric Modeling	26
2.4.4. Comments about the existing work	26
Part 2. Iterative Geometric Design	27
Chapter 3. Mathematical Background	29

3.1.	Introduction	29
3.2.	Of Monster Curves ...	29
3.2.1.	The Cantor set	29
3.2.2.	The Von Koch Curve	30
3.3.	... and Iterative Geometric Objects	31
3.3.1.	De Castejau's Method	31
3.3.2.	Iterated Function Systems	31
3.3.3.	IFS-formalism	32
3.3.4.	Projected Iterated Function Systems	33
Chapter 4.	Discrete Iterative Geometric Design	35
4.1.	Transformation driven Geometric Design	35
4.1.1.	Potential of Transformation driven Geometric Design	39
4.2.	Constrained Geometric Design	40
4.3.	New Constrained Surface Model	41
4.3.1.	Vector Sum Surfaces	41
4.3.2.	Projected Vector Sum Surfaces	42
4.3.3.	Projected IFS Vector Sum Surfaces	44
Chapter 5.	Geometric Design by Discrete Iterative Geometry	47
5.1.	Topological Aspects	47
5.1.1.	Variations on the cantor set	47
5.1.2.	Connected figures.	51
5.1.3.	Notes	52
5.2.	Curve Design	53
5.2.1.	Quadratic Bezier Curves	54
5.2.2.	Conics by Rational quadratic Bézier Curves.	55
5.2.3.	From quadratic Béziers to fractal Curves	56
5.2.4.	Cubic Bézier curves and Cubic B-Splines	60
5.3.	Orientation and Symmetry	63
5.4.	Surfaces and Operators	66
5.4.1.	Surfaces by curve products	66
5.4.2.	Multi-patch Surfaces by curve sums	68
5.4.3.	Resolution control of vector sum surfaces	70
5.5.	Discussion	72
Part 3.	Applications of Discrete Iterative Geometry to Architecture	73
Chapter 6.	From Geometry Data to Workshop Plans	75
6.1.	Work Flow: Digital Chain	75

6.2.	Addressing	75
6.2.1.	Addressing function provided by the IFS formalism	76
6.2.2.	Limitations of IFS-addressing	80
6.2.3.	Addressing faces of quadrilateral mesh figures	81
6.2.4.	Defining Neighbors Geometrically	84
6.3.	Constructional Elements	87
6.3.1.	From quad mesh faces to chamfered timber panels	87
6.3.2.	Border conditions for Border elements	89
6.3.3.	Examples of thickened planar quad meshes	91
6.3.4.	Conical Meshes	92
6.3.5.	Offset of non-conical meshes	93
6.3.6.	Notes on offset meshes	95
6.4.	Generic Detailing	95
6.4.1.	Linear fastener distribution	96
6.4.2.	Fasteners for joining timber panels	98
6.4.3.	Montage of quad panel structures	102
6.5.	Discussion	104
Chapter 7.	Integrated Manufacturing	105
7.0.1.	Scope of this chapter	105
7.0.2.	Integrated manufacturing techniques	105
7.1.	Mapping Problem	105
7.2.	Packing Problem	108
7.2.1.	Description of the goods	109
7.2.2.	Packing process	109
7.3.	From Geometry Data to Machine Instruction	112
7.3.1.	Machining conditions	112
7.3.2.	Machining strategies	113
7.3.3.	Tool path generation	115
7.3.4.	From tool-paths to G-code	119
7.4.	Testing and discussion	120
7.4.1.	Sample production	120
Chapter 8.	Applications by Examples	123
8.1.	Decor Panels	123
8.1.1.	Hilbert Panels	123
8.1.2.	Fractal Shading Panels	128
8.1.3.	Comments on ornamental applications	130
8.2.	Bézier Vault	130
8.2.1.	Generation of the manufacturing data	131
8.2.2.	Note	133

8.3. B-spline Shell	134
8.3.1. Bending planar construction material onto non-planar geometries	134
8.3.2. Perturbation method to lessen the local curvature of non-planar quad meshes	137
8.3.3. Automation of the process	139
8.4. Timber Panel Structures	141
8.4.1. Manufacturing	143
8.4.2. Prototypes	144
8.4.3. Conclusions	148
8.5. Discussion	154
Part 4. Conclusion and Outlook	155
Chapter 9. Conclusion	157
9.1. Discussion	157
9.2. Contributions	158
9.2.1. Iterative Geometric Design	158
9.2.2. Generation of Constructional Elements of free-form architectural objects	159
9.2.3. Integrated Manufacturing	160
9.3. Contributions to the State of the Art	161
Chapter 10. Outlook	163
10.1. Applications	163
10.2. Future Work	164
10.2.1. Research on Iterative Geometric Design	164
10.2.2. Development in Integrated Architectural Design	165
10.2.3. Future Applications	165
Epilogue	166
Bibliography	167
List of Figures	171
Nomenclature	177
Chapter 11. Appendix	179
11.1. About this Appendix	179
11.2. Geodesic Rib Shell	180
11.2.1. Project Description	180
11.2.2. Structural System	180

11.2.3. Parameters	182
11.2.4. Resulting Constructional Elements	183
11.2.5. Conclusion	186
11.3. Applications to Integrated Production	187
11.3.1. Festival Room Model	187
11.3.2. Production of an Origami Eye-catcher	189
11.3.3. 3-axis Machine Post-Processors	193
11.4. Load Bearing Capacities of Iteratively Designed Geometries	196
11.5. Software Implementation	197
Curriculum Vitae	199

Part 1

Introduction

CHAPTER 1

Goals and Methods

This work investigates on computer aided integrated architectural design and production. The aim is to provide integral solutions for the design and the production of geometrically complex free-form architecture. Investigations on computer aided geometric design and integrated manufacturing are carried out with equal importance. The research is considering an interdisciplinary approach, including computer science, mathematics and architecture.

The goals of this research can be considered individually with regards to three distinct fields of interest, which are:

- Geometric design
- Digital production of architecture
- Applications

Within the following, the goals concerning each of the above mentioned categories are explained more in detail. However, it is important to note that the integral and multi-disciplinary approach is of high importance. In terms of efficient design and production of geometrically complex architecture, we believe that the design process can not be optimized regarding the concerned fields on its own only. Therefore, the solutions on the level of geometric design have to consider notions of the level of production, and vice versa.

1.1. Geometric Design

1.1.1. Iterative Geometric Design. The goal in terms of geometric design is to provide a model, which is general enough to represent a large variety of shapes. The figures, which the model should be capable to construct, are:

- classical smooth figures (Béziers, Splines, NURBS etc.)
- Subdivision curves and subdivision surfaces (e.g. Cattmul-Clark)
- Fractal figures (Von Koch curve, Sierpinski triangle, Dürer pentagon etc.)
- L-Systems (Trees, plants, etc.)

A general model, which integrates the above mentioned shape families, would present the advantage that it would provide the possibility to change continuously between different shape grammars. Today, the user is forced to decide in advance whether he is going to design a smooth Bézier-surface or an accidented fractal figure. There is actually no tool which allows natural transition between these separate design paradigms.

Iterated function systems, IFS, described by Barnsley [Bar88], have been developed in order to formalize the construction of fractal objects. However, the rather general expression of the formalism might also be used for the representation of other than only fractal figures. Subdivision modeling, which is mainly used for the construction of smooth figures, can be simply represented within the scope of IFS-Modeling.

The basic idea of using IFS to generate a large number of possible shapes, aims to provide architects a geometric design method which unifies the until now separate paradigms of the smooth and the rough.

1.1.2. Control Parameters. Methods of shape control should provide both, a graphical and a declarative way to manipulate the figures designed. It may provide common interface methods known from NURBS-modeling, which are namely the control of the figure by its control points. The control points may be organized in different ways. Depending on the topology of the modeled figure we will work with control-polygons, -grids, -branching structures, to name just a few. The control points provide a intuitive graphical interface to manipulate the geometric figure at the computer screen. In addition to this, a declarative interface, which allows the input of precise coordinate data, should be provided simultaneously.

While the control points allow the manipulation of the figure at a global level, they do not offer the design of the local aspect of the geometry designed. Therefore, further control has to be established which will act locally on the shape's properties in order to allow defining respectively the roughness and the smoothness of the figure.

1.1.3. Constraints. The free-form geometries designed at the computer screen lack generally in physical feasibility. Since CAGD-software does not consider physical properties, as for example gravity, it is obvious that computer aided geometric design makes it possible to design a lot more shapes than the ones that we are actually able to build. The fact that we intend to produce feasible figures, which are not only displayed on a computer screen, forces us to restrict in some terms the liberty of the designer. Criteria, that define what is constructible, have to be identified.

These criteria will further be translated into a list of constraints, which the geometry engine must verify, in order to provide figures suitable for physical realization. While the definition of the topological and geometrical constraints is rather straight forward, the integration of them into the geometric design method turns out to be rather difficult.

Within this thesis, only some constraints are going to be considered. First, question related to topological constraints will be treated. Later, specific geometrical constraints are considered. The aim of the constraints is to describe a system which gives control over following properties:

- Connectivity vs. discontinuity
- Overlay and self-intersection
- Control of the local structure
- Planarity of the components

1.2. Digital Production of Architecture

The conversion of purely geometric figures into applications in the field of architecture poses a certain number of questions. As stated earlier in this chapter, the virtually designed geometries do not correspond to the constraints emanating from the different construction processes in the field of architecture. The designed geometric objects, are often locally double curved, what makes their physical realization difficult. The production of double curved elements is time intensive and its costs remain relatively expensive - despite of the use of integrated manufacturing techniques. A certain number of points aiming to optimize the production costs, is already integrated in the constraints which the geometric model verifies. Further optimization tries to close the gap between the early design stage at the computer screen and the physical realization in the carpenter studio.

Generally, the geometric object does not meet all properties that are necessary for its physical construction. Therefore, several post-processing methods will have to be applied in order to translate the geometric data into a coherent set of constructional elements. These post-processing methods can be split into the following work steps:

- Computation of the geometric data of the constructional elements
- Mapping functions relating data from the geometric figure to the constructional elements

- Automated creation of machine instructions for integrated manufacturing
- Addressing and labeling of the constructional elements for facilitating the logistics during both production and assembly

1.3. Applications of Discrete Geometry to Architecture

The application of iterative geometry for architectural use is central to this work. It provides a way to verify the validity of the established methods. The range of potential applications reaches from purely decorative architectural elements, representing an ornamental character, to big scale structures, which provide load bearing capacities. The applications are tested on a series of reduced scale prototypes, which verify the suitability of iterative geometric design for architectural use.

CHAPTER 2

State of the Art

This chapter provides an introduction into the topic of geometric design for architectural use, the broader framework of this research as well as recent development in the fields of timber constructions and fractal geometry.

2.1. Architectural Geometry

2.1.1. On Architectural Geometry. Since the early nineties, architects have shown greater interest in the use of computer aided design tools. Nowadays, using computers and computer aided solutions for the design and the production of complex architectural projects has become fairly common. The new computer aided design tools have basically been developed by the automotive and aeronautic industry. Some tools have been created by the industrial design or the motion picture industry.

The software solutions, which today's architects work with, provide a large variety of very powerful tools for the generation of non-classical geometric figures. Common geometric design methods provided by modern CAD-software are NURBS and Subdivision modeling techniques. The tremendous variety of shapes, which might be easily generated by such tools, has to be considered relatively to the actual ability to physically produce the buildings. The production methods employed in the field of architecture and the properties of architectural design differ greatly from the range of applications the computer aided tools have been developed for initially. Until today, physical realization of free-form architectural objects remains rather difficult and demands tremendous effort in terms of planning time and production cost.

In the last decade, numerous architects and engineers have tried to face the problem of lacking feasibility. Several specialized solutions for

particular projects have been worked out. In some cases, major architectural offices have founded separate departments for researching on dedicated solutions for the realizations of geometrically “complex” objects¹. The fabrication techniques in the automotive industry use mass production. Contrary to this, the manufacturing techniques for the production of architecture is oriented for the production of prototypes. In architecture, a particular design is generally built only once, whereas a specific car model is designed to be produced in series of ten-thousands or more.

The fact that each building is in some way a prototype makes difficult the development and optimization of technical solutions with regards to rational production of complex architecture. This may partially explain why there is no specialized software, as for car design, which integrates criteria from the production at an early design stage. In the field of architecture, we have to deal with the problem of mass-customization, which is far different from working with the concept of mass-production.

The problems addressed within the previous paragraph let several scientists investigate on the problem of architectural geometry. In order to provide new geometric methods for the efficient realization of geometrically complex architectural objects, tremendous effort is undertaken. Most of the time, the discovered solution is focusing on a very particular problem. In the field of discrete geometry, a lot of interesting work has recently been done [PBCW07].

2.1.2. Mathematical Models for Architectural Use. The architect employs principles of geometry in an applied way. The mathematical methods, more precisely geometrical methods, serve as tools which help to develop, refine and communicate his designs. Several historical examples demonstrate the interest in applied architectural geometry. Below, a selective list presents scientists endorsing the use of geometrical methods as tools for the architect’s profession:

- (1) Vitruvius insists in [PoIBC] about the importance of geometry and mathematics as a tool for the architect. In his ten books, he teaches the work of the mathematicians Pythagoras, Archytas and Eratosthenes as base for the representation of the architectural design (ichnographia, orthographia, scaenographia).

¹The term complex geometry is often employed in the field of construction for buildings which were not designed using the principles of euclidean geometry. Modern geometric design methods offered by actual CAD-software include NURBS-modeling techniques, to which the designing architect is not specially familiar. The lack of mathematical knowledge of the architect does not mean in any sens that such geometric figures are complex in a mathematical sense.

These mathematical methods provide rules for proportion, harmony and order (*venustas*) and may be used for precise cartography. In his opinion, the architect's multidisciplinary profession incorporating large knowledge in different domains is the most excellent of all sciences ("sumum templum architecturae").

- (2) Some of Dürer's most famous drawings are teaching the principles of the central perspective based on Alberti's method, which became a common architect's tool for the representation of designs. Besides this, Dürer publishes in "Four Books on Measurement" (cf. [Dür38]) variations of Platonic solids by studies on Archimedes's semi-regular solids. Knowing the work of Vitruvius, he talks about Euclid's Elements, Ptolemy and Apollonius, who taught him several methods of curve design. Finally, in the third book, the principles of geometry are applied to architecture and civil engineering.
- (3) The french mathematician Gaspar Monge founded the principles of descriptive geometry. In addition to his work on analytical geometry [Mon50], his work on descriptive geometry has been published in [Mon47]. After a couple of decades of non-disclosure for military use, Monge's method of descriptive geometry became available to architects and civil engineers around 1850. Until today, his method is used as a standard for technical drawings.

The wide use of descriptive geometry in architecture shows the importance of applied mathematics in the field of architectural design. However, using mathematical and geometrical methods for the architectural design demands a found knowledge of the employed techniques. Recently, several new geometric design methods - namely, computer aided geometric design methods - have become available to the architect. These new computer aided design tools may be considered easy to use since they are provided with intuitive modeling techniques offering interactive graphical user interfaces. Too often, the designer behind the computer screen does not have an accurate understanding about the implied geometrical methods, which can be limiting the potential of the powerful design tools offered.

2.2. New Methods in Timber Construction

Working in the field of timber construction, this research is focused on application within that domain. Even though the geometric design methods and some of the production techniques treated are not exclusively restricted to the field of timber construction. It is important to note that recent technological changes in timber construction have particularly motivated us to focus our research on timber for the design and construction of free-form architecture.

2.2.1. Integrated Manufacturing. Recently, the industry of timber construction in central Europe has seen important changes. The technological changes can be considered as a second industrialization in this domain. Large scale integrated manufacturing devices have been installed in all major carpenter studios. The manufacturing process has mainly been automated. Several products, such as Hundegger or LignoCam, allow efficient realization of standardized cuts. The software used allows rapid realization of complicated wood-wood connections. The importance of the investments in this sector witness of a general interest in new technologies.

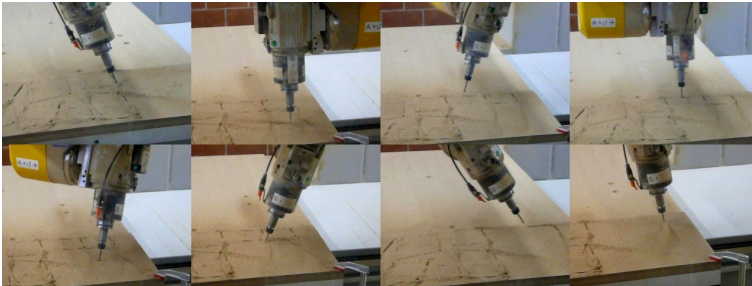


FIGURE 2.2.1. Example of a 5-Axis milling machine

These new technologies are mainly used to produce established and traditional constructional systems. The potential of the installed equipment is far from being exhausted. 5-Axis cutting machines (cf. figure 2.2.1) can produce virtually any constructional part. They are generally multipurpose devices that can be easily adapted to more specific tasks. The production cost of a particular part is mainly depending on the machining time. In terms of manufacturing time, there is absolutely no difference in machining a constructional element of a free-form geometry compared to a standardized element, regardless of the geometry of the

part being produced. The difference lies in the programming of the machine, or more precisely, in the generation of the machine instructions. While there exists tremendous experience for the production of standardized constructional elements, the generation of the machine's instructions for the production of "complex" geometry remains too often done by hand, which means high manufacturing costs.

2.2.2. New Timber Products. Within the last few years, numerous innovative timber products have been developed. Among the diversity of new products on the market, our interest relies particularly in big size derived timber panels. The composition of the panels vary from one product to another. There exist glued laminated block panels or laminated panels with nailed, screwed or doweled connections. A more detailed overview of the existing products can be found in [BW07]. The main interest of these panels is their size, which allows using them as wall or slab element. This is also their main field of application. Another major advantage of these timber derived products remains in their suitability to be used for structural applications. The use of timber block panels for load bearing walls has led to specific products, which differ from the ones used for slabs.

The above mentioned panels are derived timber products providing a higher homogeneity than the natural material wood. This homogeneity allows to employ the material wood in new and unseen ways. Combined with integrated manufacturing, new applications may be developed providing solutions to the production of geometrically complex free-form architecture.

2.2.3. New Timber Structures. Given the variety of new timber products and the disposal of new manufacturing methods, research on new and innovative timber structure has been carried out recently. At the laboratory for timber construction, IBOIS, professor Yves Weinand's team is researching on different new types of timber structures dealing with new geometrical design methods. The research projects are dealing with Origami structures [BW06b, HW07, HW08b, HW08a, BW08b], woven structures [HW09], fractal structures [SGW08, TBSG⁺06, TSGW07, GTSW08, SWG08, GTSW09] and free-form grid shells [WP06, PW06, NW, NW07]. A summary of the ongoing research field is resumed in [Wei08a, Wei08b].

2.3. Broader Framework of the Research

This thesis is part of a broader research project entitled “Fractal Geometry and its Applications in Timber Construction” which started back in 2005. The project team consists in a group of scientists working in the field of architecture, mathematics and computer science. The goal of this interdisciplinary research team is to develop computer aided solutions for the realization of geometrically complex architectural objects.

Closer study of fractal modeling showed that the underlying rule-set, which allows the construction of highly complex fractal figures, is rather simple. The simplicity of the mathematical method used for the construction of fractal shapes drove the team to put up following hypothesis: The method of iterative fractal modeling will allow optimizing the construction of “free-form” architecture. This hypothesis assumes that the fact of using simple rules for the construction of geometrical figures will simplify the construction of the physical architectural object. The geometric parts, which constitute fractal figures, are all treated equally. This lets assume that the constructional elements building up the final architectural object will be rationalized, if designed by fractal iterative modeling. This rationalization will allow a more efficient handling of the geometric data which in term will allow automated generation of manufacturing plans and finally will permit integrated production of the constructional parts.

On the one hand, fractal modeling showed potential in terms of optimization. On the other hand, it offered a large variety of shapes obtainable by iterative modeling.

2.3.1. Interdisciplinary Research Team. The research mentioned above was carried out by professor Yves Weinand, civil engineer and architect, professor Peter Buser, mathematician working in the field of geometry and topology, Dr. Eric Tosan, mathematician and computer scientist working in the field of fractal geometry.

The possibility to work within that interdisciplinary setting provided tremendously large knowledge and rich moments of discussion and exchange. Initially, three PhD. candidates working in the same laboratory were forming of the team. Being architect, notions of geometric design and software development had to be understood and considered. The established dialog enabled each other to enlarge his individual knowledge and to integrate concerns of the other domains.

2.4. Existing Work in Fractal Geometry

2.4.1. Applications of Fractal Geometry. Since about 1970, fractal geometry as a new branch of mathematics influences the development of science in various fields (physics, chemistry, medicine etc.). In addition, its mathematical methods influenced visualization software. Here are some of the domains in which contemporary fractal geometry is applied:

- 3D virtual imaging (mountains, plants, clouds, etc.)
- physics, chemistry, physiology
- digital image compression (JPEG)
- technologies (fractal antennas)

The following are standard references dealing with fractal geometry in a general way: Falconer [**Fal90**], Barnsley [**Bar88**] and Prusinkiewicz [**Pru89**]. They are considered as sources which serve indirectly the purpose of our research.

At the origin of fractal geometry, one consideration was the geometric construction of natural objects. The best known reference being Mandelbrot [**Man82**]. Mandelbrot discovered the method of fractal geometry initially to describe the shape of natural objects. In contrast to this, our research does not focus on the use of this method in order to imitate natural objects but rather uses the formalism of fractal geometry to create architectural and design objects. More precisely, we use a formalism inspired by IFS [**Bar88**] or L-system [**Pru89**].

2.4.2. Fractals in Architecture. The use of fractals in architecture at the scale of a city has been studied by Humpert [**HH89**] and Batty [**BL94**] who analyze the city by means of different fractal aspects, which are: self-similarity in scale (of the city's texture), fractal growth (looking at the overall shape in different times) and fractal size distribution (dispersion and aggregation of urban agglomerations). To understand fractal size distribution of buildings in settled areas one may consult Frankhauser [**Fra94, Fra97**]. For the concept of self-similarity of the shape of a building at different scales we refer to Trivedi [**Tri89**] and Sala [**Sal02**]. The phenomena of self-similarity in patterns of ornamental use can be studied within the work of Bovill [**Bov03**] and Rodriguez [**RR03**]. The notion of self-similarity in scale will be considered in our research. Investigations in the field of urban design are not considered within the scope of the presented study.

In the field of civil engineering, a few papers exist, dealing with applications of fractal geometry (for example the papers on fractal FEM-meshes [**LDFS04, MP02**]).

2.4.3. Fractal Geometric Modeling. Recently, different research projects piloted by the LIRIS have been studying fractal geometric modeling. Within this work, different methods have been developed for extending classical geometric design by fractal shapes. It has been shown in [**ZT96, TZTV97**] that it is possible to manipulate fractal figures with the same methods as the ones used for classical geometric design, as i.e.: control point editing, deformations or tensor products. Further, the introduction of constraints on the transformations matrices provides control on some topological properties of the generated figures [**Tos99, TGB02**]. Hereby, a general modeling method (IFS), which has initially been developed for the visualization of natural objects, has become a geometric design method producing well structured figures (i.e.: face-edge-vertex meshes). The research on fractal geometric modeling has extensively been published in [**Gen92, Tho96, Tos96, Zai98**].

2.4.4. Comments about the existing work. Even though fractal geometry has seen different industrial applications, to our knowledge, there exist no direct applications of fractal geometric design for the creation and the production of free-form architecture. In the field of architecture, several works have been carried out (cf. 2.4.2) using fractals for the analysis of architectural patterns at different scales. The use of fractal geometry as a design tool represents a completely new approach. Therefore, the stated references are seen as background and do not directly show an influence on the present research. However, we will use and develop further the IFS-formalism developed by Barnsley [**Bar88**] and by Tosan et al. [**ZT96, TZTV97**]. This will be addressed within the following chapter.

Part 2

Iterative Geometric Design

Mathematical Background

3.1. Introduction

In order to present the geometric design method studied, some mathematical background needs to be explained. Before explaining the principles of transformation driven geometric design, a series of historical examples will be discussed. This will introduce the reader to the methods of iterative geometric design. The relation between the mathematical method of geometric design and the physically constructed building will be shown by examples in part 3.

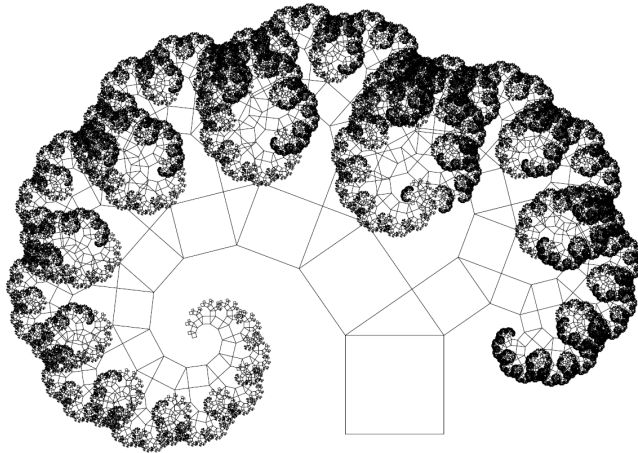


FIGURE 3.1.1. Pythagoras tree

3.2. Of Monster Curves ...

3.2.1. The Cantor set. The Cantor set, also called Cantor dust, is named after the German mathematician Georg Cantor [Can84]. It describes a set of points which lies on a straight line. In the end of the 19th century, this figure attracted the attention of mathematicians because of

its apparently contradictory properties. Cantor himself described it as a perfect set, which is nowhere dense. Further properties, such as self-similarity, compactness and discontinuity, have been studied years later. The geometrical construction of the Cantor set can be explained as follows: Take a straight line segment, divide it into three parts of equal length and remove its middle third; divide again each of the resulting line segments and keep removing their middle thirds. If you repeat this for each of the new line segments, you will end up with the Cantor set (cf. figure 3.2.1).

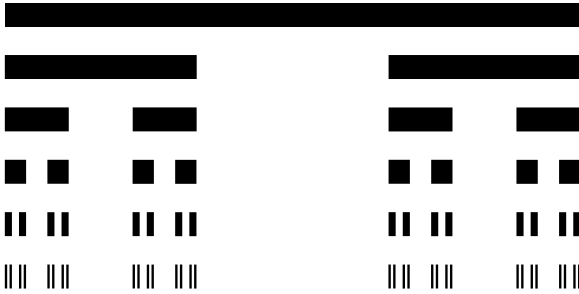


FIGURE 3.2.1. Cantor set

3.2.2. The Von Koch Curve. The Von Koch curve belongs among the first found and best known fractal objects. In 1904, the Swedish mathematician Helge Von Koch described it for the first time in [Koc04]. The Curve is constructed stepwise. Beginning from a straight line, there results a meandering curve (cf. figure 3.2.2) with following strange properties:

- It does not possess a gradient, which means that it can not be differentiated
- The length of any of its sections is always infinite

The geometrical construction of the Von Koch curve is iterative, where each of the construction steps consists of four affine geometric transformations. The primitive is a section of a straight line, which is scaled, rotated and displaced by each of the transformations $\{T_1 \dots T_4\}$. Per construction step, four duplicates are generated, of which each will produce four more duplicates in the next construction step.

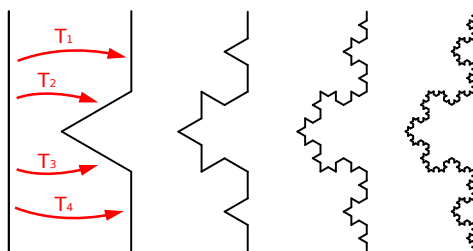


FIGURE 3.2.2. Von Koch curve

3.3. ... and Iterative Geometric Objects

3.3.1. De Casteljau's Method. In order to top off this series of introductory examples, we would like to address briefly the Bézier curve. In 1959, De Casteljau discovered an algorithmic method for the construction of the today called the Bézier curve (cf. figure 3.3.1). De Casteljau was employed by Citroën for the development of mathematical methods describing the geometry of the car's autobodies. De Casteljau's method is based on iterative construction [dC59, dC63], which is highly similar to the construction of a Von Koch curve. The actual Bézier curve was analytically described by Bézier in 1962 as a polynomial function [Béz62]. Simultaneously to the work of Paul De Casteljau, Pierre Bézier was working at Renaults and developed independently from his concurrent de Casteljau the software Unisurf, which presents the headstone of today's computer aided geometric design software - the concepts of CAD and CAM were born.

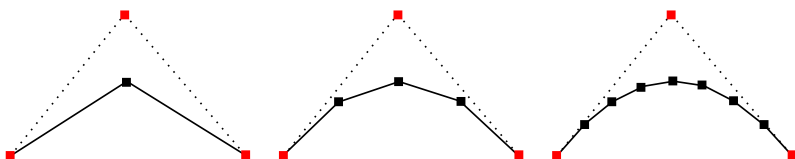


FIGURE 3.3.1. Iterative construction of a Bézier-curve

3.3.2. Iterated Function Systems. The strange properties of the aforementioned objects led the mathematicians to name them “monster curves”. In 1981, based on Hutchinson's operator [Hut81], Barnsley defined a formalism which was able to describe such objects in a deterministic way [Bar88]. His IFS-method (Iterated Function Systems) consists in a set of

contracting functions that are applied iteratively. In our case, a function is an affine geometric transformation. Iterative means that the construction is done step by step. The input of a construction step is the result of the step before. What is really new in Barnsley's detection is that the resulting geometric figures are not defined by the primitive used, but rather by its transformations. The proof is the construction of a Sierpinski triangle, which uses a fish as primitive. Analogous to this, the Von Koch curve might be constructed based on the letter "A". The result we end up with remains strictly the same. The conclusion, that it is theoretically possible to use any form of primitive for the construction of such geometric figures, led us to the hypothesis that it is basically possible to use construction elements as primitives. Instead of using fishes like Barnsley, we would rather use construction elements such as beams or panels etc.

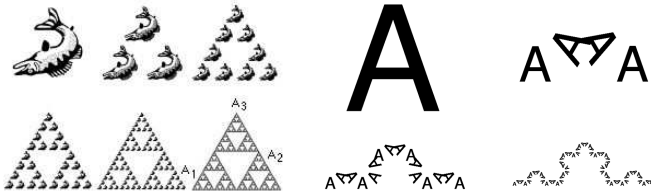


FIGURE 3.3.2. Barnsley's illustration for IFS

3.3.3. IFS-formalism. The geometric figures of the examples shown in the beginning of this section are all defined by a set of transformations $\{T_i\}$. As Barnsley teaches us, the result is indifferent to the initial object on which is applied the set of transformations. This is true for the limit state, also called attractor A , which is the resulting figure if the set of transformation $\{T_i\}$ is applied an infinite number of times. The attractor A is the unique non-empty compact such that

$$A = \bigcup_i T_i A$$

The attractor, as shown in figure 3.3.3, is a theoretical object, which we will not discuss any further. We will work on the intermediate construction steps that are given by the following sequence:

$$K_{n+1} = \bigcup_i T_i K_n$$

Where K represents the initial figure, the so called germ. K might be any arbitrary object as for example a fish, as shown in figure 3.3.2, or a

straight line segment, as used for the construction of a von Koch curve or a Bézier curve (cf. figures 3.2.2 and 3.3.1).

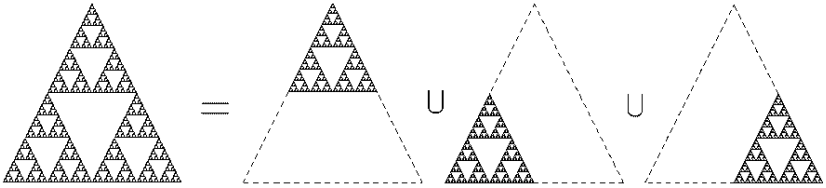


FIGURE 3.3.3. Graphical representations of the Attractor as a union of its components

Finally, the following equation shall be mentioned, which relates the geometric figure K at construction step $n \rightarrow \infty$ to the attractor A :

$$\lim_{n \rightarrow \infty} (K_n) = A$$

Within the following, the modeled objects will be defined as the projection of an IFS via a projection operator P .

3.3.4. Projected Iterated Function Systems. While the IFS formalism presented in 3.3.3 is fully defined by the parameters of the transformations $\{T_i\}$ and the initial primitive K , the projected IFS integrate control points P as further parameters into that formalism. Hereby, figures created by projected IFS can be controlled in a similar way as classical geometric figures such as for example Bézier curves or Splines (cf. figure 3.3.4).

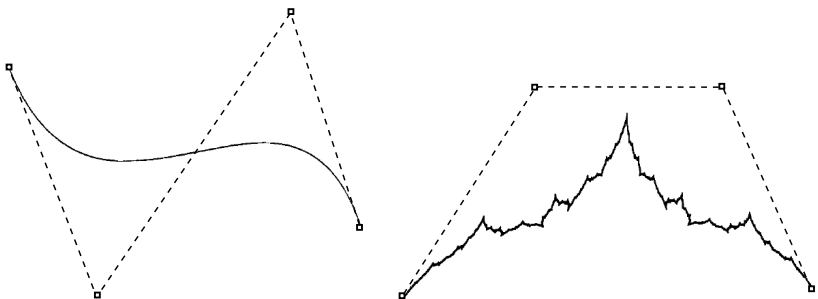


FIGURE 3.3.4. Smooth and fractal figure with control points designed by projected IFS according to [ZT96]

The design of iteratively constructed curves by control points has first been discovered by De Casteljaeu, who used it for the construction of smooth figures (cf. section 3.3.1). Overveld, Prautzsch and Michelli studied recursively defined curves related to cubic Bézier curves in [**vO90**, **MP87**, **PM87**], where the De Casteljaeu's method was partially extended to fractal like curves ("wild curves"). However, the control of fractal objects by the aid of control points was first proposed by Tosan and Zair in [**ZT96**, **Zai98**], who integrated the control points P into the formalism of IFS as a projection operator.

Discrete Iterative Geometric Design

The strange properties of the geometric figures discussed in section 3.2 are concerning the limit state, the attractor. For practical applications, the theoretical object of the attractor A is far less relevant than its intermediate construction state K_n . In the scope of this work, the construction state K_n has following properties:

- K_n is computational point by point
- The resulting geometry is always expressed by a finite number of elements (points, lines, faces)

While the above mentioned properties of the figures are suitable for their geometric construction, they present also an interest suitable for software implementation of the design method.

Further, the fact that the geometry data presents a discrete expression is advantageous for later physical realization. Discrete geometry presents certain advantages for the application of free-form geometries to architecture such as the ones discussed later on in part 3 of the present document.

4.1. Transformation driven Geometric Design

As stated in section 3.3.3, the final aspect of the iteratively constructed geometry is defined by the transformations. In order to control the resulting figure, the geometric design method has to provide solutions, which allows acting on the transformations used for the construction of the figure.

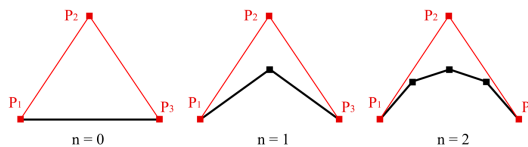


FIGURE 4.1.1. First construction steps of a Bézier curve

An example of an iteratively constructed Bézier curve is shown in figure 4.1.1. The figure shows the first construction steps of a Bézier curve with three control points P shown in red.

$$P = (p_1, p_2, p_3) = \begin{pmatrix} p_{1,x} & p_{2,x} & p_{3,x} \\ p_{1,y} & p_{2,y} & p_{3,y} \\ p_{1,z} & p_{2,z} & p_{3,z} \end{pmatrix}$$

At the beginning ($n = 0$), the initial figure K consists in the line connecting the end points of the control polygon $[p_1, p_3]$. Per construction step, two transformations $\{T_1, T_2\}$ are applied on each element. Below, the first three steps of the construction sequence are shown.

$$\begin{aligned} PK_0 &= PK \\ PK_1 &= PT_1K_0 \cup PT_2K_0 = PT_1K \cup PT_2K \\ PK_2 &= PT_1K_1 \cup PT_2K_1 = PT_1T_1K \cup PT_1T_2K \cup PT_2T_1K \cup PT_2T_2K \\ PK_3 &= PT_1K_2 \cup PT_2K_2 = PT_1T_1T_1K \cup \dots \cup PT_2T_2T_2K \end{aligned}$$

Each element of PK_i can be computed by following the construction tree shown in figure 4.1.2.

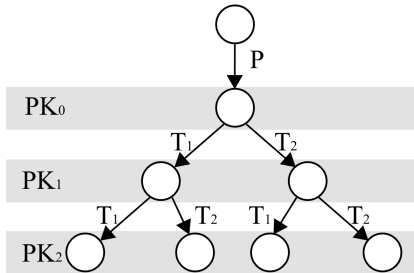


FIGURE 4.1.2. Construction sequence: Tree representation

For our work, transformation matrices are used to describe affine geometric transformations. We work with barycentric coordinates, which means that each computed point is based on a combination of the entry points. In the present example of a Bézier curve with three control points, each transformation can be expressed by one 3x3 matrix:

$$T_1 = (c_1, c_2, c_3) = \begin{pmatrix} 1 & 0.5 & 0.25 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 0.25 \end{pmatrix}$$

$$T_2 = (c_4, c_5, c_6) = \begin{pmatrix} 0.25 & 0 & 0 \\ 0.5 & 0.5 & 0 \\ 0.25 & 0.5 & 1 \end{pmatrix}$$

In the field of subdivision modeling, the above shown transformation matrices are rather standard. They work with fixed values conducting always to the same result: a smooth Bézier curve with three control points.

A closer look at the transformation matrices of the Bézier curve shows that they have certain values in common. For example: The column c_1 of T_1 as well as the last column c_6 of T_2 is fixed. Further, the last column of T_1 is identical to the first column of T_2 . These dependencies guarantee that the resulting curve will be *continuous*. The rest of the values are free and can be modified at the designer's desire.

If we want to generalize the above mentioned findings, we can say that certain values of the transformation matrices are constrained whereas others are free. Based on the example of an iteratively constructed continuous curve with two transformations and three control points, the general scheme of the transformation matrices is:

$$T_1 = \begin{pmatrix} 1 & c_{2,1} & c_{3,1} \\ 0 & c_{2,2} & c_{3,2} \\ 0 & c_{2,3} & c_{3,3} \end{pmatrix}, T_2 = \begin{pmatrix} c_{3,1} & c_{5,1} & 0 \\ c_{3,2} & c_{5,2} & 0 \\ c_{3,3} & c_{5,3} & 1 \end{pmatrix}$$

In order to act on the free values of the transformation matrices, we define a graphical data input method. We introduce the set of points $\{s_i\}$, which we call "subdivision points" (cf. figure 4.1.3).

$$(s_1, s_2, s_3, s_4, s_5, s_6) = \begin{pmatrix} s_{1,x} & s_{2,x} & s_{3,x} & s_{4,x} & s_{5,x} & s_{6,x} \\ s_{1,y} & s_{2,y} & s_{3,y} & s_{4,y} & s_{5,y} & s_{6,y} \\ s_{1,z} & s_{2,z} & s_{3,z} & s_{4,z} & s_{5,z} & s_{6,z} \end{pmatrix}$$

Each subdivision point s_i corresponds to the projection via P of a column c_i . In other words, the subdivision point s_i is the image of the control points P by c_i . Therefore:

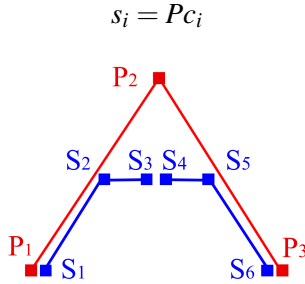


FIGURE 4.1.3. Representation of control and subdivision points

The relation $s_i = P c_i$ between $\{p_i, s_i, T_i\}$ is graphically illustrated by figure 4.1.4.

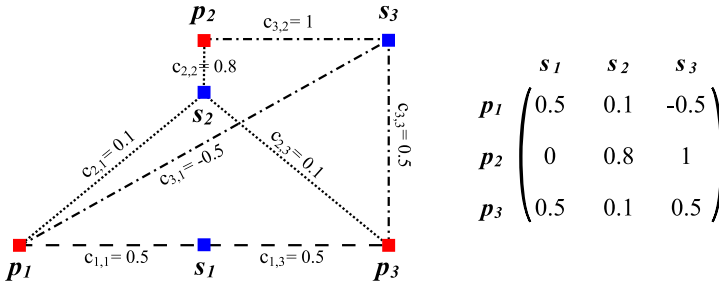


FIGURE 4.1.4. Relation between the control points P , the subdivision points S and the transformations matrices T

Accordingly , the coordinates of the subdivision point s_i are obtained by the control points P transformed by c_i as follows:

$$s_i = \begin{pmatrix} s_{i,x} \\ s_{i,y} \\ s_{i,z} \end{pmatrix} = \begin{pmatrix} p_{1,x} \cdot c_{i,1} + p_{2,x} \cdot c_{i,2} + p_{3,x} \cdot c_{i,3} \\ p_{1,y} \cdot c_{i,1} + p_{2,y} \cdot c_{i,2} + p_{3,y} \cdot c_{i,3} \\ p_{1,z} \cdot c_{i,1} + p_{2,z} \cdot c_{i,2} + p_{3,z} \cdot c_{i,3} \end{pmatrix}$$

Note: In the example shown in figure 4.1.4, the subdivision point s_1 depend only on $\{p_1, p_2\}$ while the subdivision points $\{s_2, s_3\}$ are depending on a combination of all three control points $\{p_1, p_2, p_3\}$.

By this relation, the values of the columns c_i can be obtained by the position of the subdivision points s_i relative to the position of the control points P , using the equation below:

$$c_i = P^{-1}s_i$$

where P^{-1} is the inverse matrix of the control points P .

This property allows us to act on the transformation matrices T by manipulating the subdivision points. Figure 4.1.5 shows two cases of IFS-figures where the coordinates of the subdivision points $\{s_i\}$ have been modified relatively to the control points P . Effectively, the values $\{c_{i,i}\}$ of the subdivision matrices $\{T_i\}$ have been changed accordingly, conducting to two different figures: A smooth Bézier curve, on the left, and a rough fractal curve, on the right.

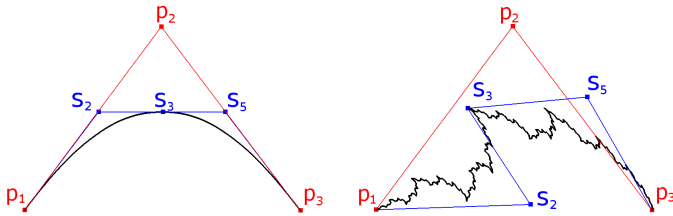


FIGURE 4.1.5. Acting on the transformation matrices by manipulating the subdivision points (shown in blue)

4.1.1. Potential of Transformation driven Geometric Design. Classical subdivision schemes allow the creation of smooth shapes. Their entry parameter is a set of control points, each of which can be freely positioned in the model-space (generally \mathbb{R}^3). The data of these control points are used as bases for the subdivision algorithm, which utilizes predefined matrices to compute the new points. Contrary to this, the subdivision matrices in our model are not pre-defined but can be parametrized by S . These new parameters are graphically represented by points, which can be freely positioned in the model space in the same way as the control points. Each of these points - which we called subdivision points S - is the image of the control points P by the subdivision operator.

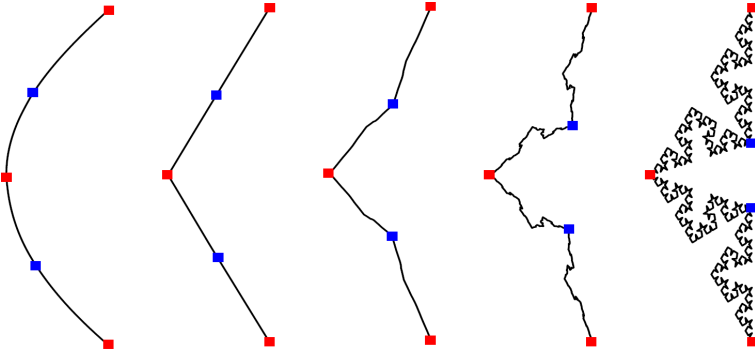


FIGURE 4.1.6. IFS curve design: adjustment of roughness / smoothness

This gives us the possibility to act on the local aspect of the figure being designed. Whether a figure is smooth or rough depends only on the affine geometric transformations. The same curve might be smooth or rough. By changing the subdivision parameters, respectively the smoothness and the roughness can be adjusted, as shown in figure 4.1.6. The input of the subdivision parameters is controlled by the position of the subdivision points. Alongside the control points, which are widely known in classical CAD-software, subdivision points augment the variety of design possibilities. They provide a graphical way to manipulate the affine geometric transformations (subdivision operator), which are expressed in the user-unfriendly form of n -dimensional matrices that work under the skin of the graphical user interface.

4.2. Constrained Geometric Design

The goal is to develop design strategies that make the design and the production of free-form surfaces easier. Therefore, the geometric design should meet certain topological and geometrical constraints. The constraints are mainly dictated by physical and production conditions, coming from the field of construction. Within the following, a few examples of different constraints are presented.

E.g. an important point might be that the free-form object will be built out of planar timber panels. According to this, the geometrical constraint demands that the virtual 3D-model must be constituted completely of planar parts. We will work on this constraint in sections 4.3 and 8.4.

In chapter 3.2, we have presented a series of iteratively constructed objects. Not all of them are suitable for physical realization. The Cantor

set for instance, is just a set of discontinued line fractions. Since we generally need material continuity (unless designing ornaments or such), the elements building up the geometric figure have to verify that they are connected with each other. Continuity represents a topological constraint.

In order to avoid complex detailing of the nodes of a wire frame structure, it is advantageous to know the number of bars coinciding in one node. To fix the number of bars per node to 6, we might work with surfaces which are entirely composed of regular triangular faces. This is a topological constraint. On the one hand, constraints will make the physical realization of free-form objects easier. On the other hand, they may limit the design possibilities, and therefore restrict the form-finding process, what we want to avoid as far as possible.

4.3. New Constrained Surface Model

Within the following, a new surface method will be presented, which has been developed together with the computer scientist Gilles Gouaty within the scope of this work. It has been developed to provide a free-form design method which meets the requirement to produce discrete surfaces that are entirely composed out of planar elements. This method has been presented in [GTSW08] and [GTSW09].

The surfaces created by this method will be expressed in form of a discrete mesh M , which is composed of quadrilateral faces only. One problem of quadrangular faces is that the four vertices defining the face are not necessarily coplanar. The method studied, verifies the coplanarity of the four vertices defining one face. Therefore, the surface method is considered being constrained geometrically. The method uses two curves to compute the mesh M .

4.3.1. Vector Sum Surfaces. In order to create iterative surfaces, which are entirely composed of planar elements, we will work on so-called vector sums. Generally, classical CAD-software computes NURBS-surfaces by tensor products (cf. 5.4.1), which have the unsuitable property of being composed locally of double curved faces¹. Great effort is needed for their production.

The principle of using vector sums, more precisely Minkowski sums [Min96], for the generation of free-form surfaces has already been studied by Schlaich [SS02] and Glymph [GSC⁺04].

¹In this context, double curved faces are quadrilateral faces with four vertices that are not coplanar

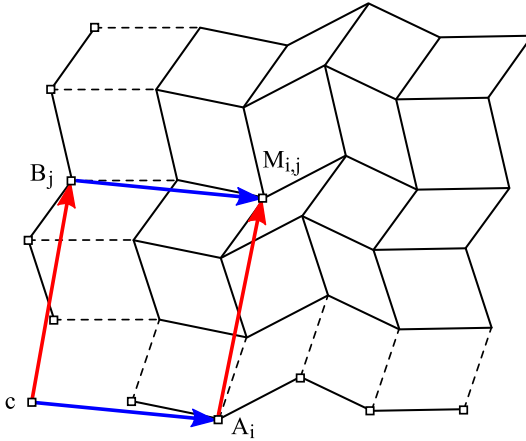


FIGURE 4.3.1. Centered vector sum surface

Vector sum surfaces are combinations of two curves. Figure 4.3.1 shows the two discrete curves a and b , which are defined by a list of points $\{A_i\}$ and $\{B_j\}$ respectively.

The resulting surface is represented by the quad mesh M , which is obtained by a vector sum of $\{a, b\}$ that is centered in point c defined by following expression:

$$M_{i,j} = c + \overrightarrow{cA_i} + \overrightarrow{cB_j}$$

Note: c is the reference point which is used to compute the vertices of the quad mesh. It may be any arbitrary point in the design space.

The surface M is completely composed of parallelograms and therefore it meets the geometrical constraint which requires that all its parts must be planar. Due to this constraint, the design possibilities of vector sums are limited compared to surfaces created by tensor products (cf. 5.4.1). In order to augment the design capabilities, Schlaich [SS02] and Glymph [GSC⁺04] extended the method such that the resulting quad mesh is composed not only of parallelograms but also of trapezoids.

4.3.2. Projected Vector Sum Surfaces. Working in \mathbb{R}^3 , the discrete curves are defined by three dimensional points i.e. $A_i = (A_{i,x}, A_{i,y}, A_{i,z})^T$ using Cartesian coordinates. Working in \mathbb{R}^3 , the surface method proposed above we will always produce quad-meshes M composed of parallelograms only. Since, the proposed method may operate on points of any dimension we propose to compute M in \mathbb{R}^4 using homogeneous coordinates. The use of homogeneous coordinates allows to extend the method

by additional parameters such as the weight of certain points. The resulting figure M in \mathbb{R}^4 is finally projected to the design space \mathbb{R}^3 . This is why the method is called projective vector sum surfaces. The projection of $M \subset \mathbb{R}^4$ on the hyper plane $w = 1$ is centered in $O \in \mathbb{R}^4$. Hereby, \mathbb{R}^3 is defined as the sub-space $W = 1$ of \mathbb{R}^4 .

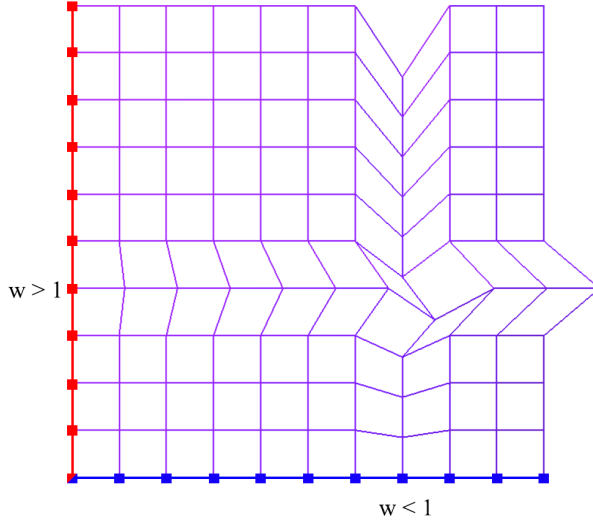


FIGURE 4.3.2. Deformation of projected vector sum meshes by editing the weight of certain points

In order to work in \mathbb{R}^4 , each point A_i will have to be defined by four homogeneous coordinates such that $A_i = (A_{i,W}, A_{i,X}, A_{i,Y}, A_{i,Z})^T$ and $B_i = (B_{i,W}, B_{i,X}, B_{i,Y}, B_{i,Z})^T$ respectively. Assigning different weights W to the points allows deforming the projected quad mesh more or less locally. Hereby, the resulting projected quad mesh M in \mathbb{R}^3 is no longer composed of parallelograms only but of convex planar quadrilaterals. To illustrate the effect of point weight editing on projective vector sums, figure 4.3.2 shows the simple case of a regular quad mesh where two point weights have been edited ($W \neq 1$).

Within the scope of projective geometry, the coordinates $(W, Y, X, Z)^T$ are called homogeneous coordinates. A point $(x, y, z)^T$ in the model space \mathbb{R}^3 defined by Cartesian coordinates with a weight W assigned has corresponding homogeneous coordinates that are $(W, X, Y, Z)^T = (W, x \cdot W, y \cdot W, z \cdot W)^T$. Reciprocally, each point of homogeneous coordinates $(W, X, Y, Z)^T$ has projected \mathbb{R}^3 coordinates which are $(x, y, z)^T = (\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W})^T$, if $W \neq 0$.

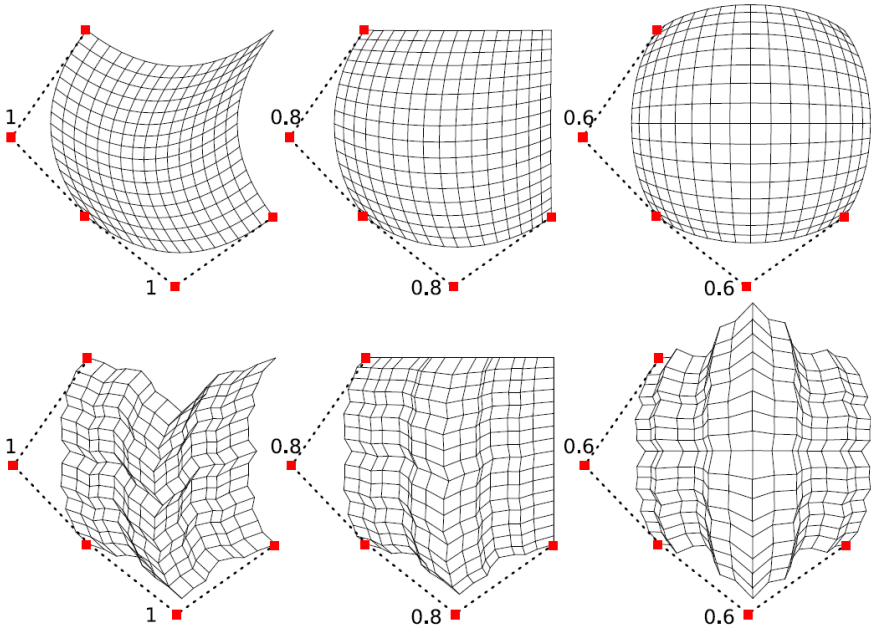


FIGURE 4.3.3. Point weight editing of smooth and rough projected IFS vector sum surfaces

4.3.3. Projected IFS Vector Sum Surfaces. The proposed surface method works with any pair of discrete curves. However, the use of IFS-curves allows controlling the global shape of the surface via its control points. In addition to that, the subdivision points allow acting on the local aspect of the surface: smoothing / roughening. In order to work with IFS-subdivision in homogeneous coordinates, the control points P and the subdivision points S must be defined in homogeneous coordinates. Hereon, the IFS-subdivision can be done in the same way as if working with Cartesian coordinates.

Examples of point weight editing on IFS-curves are presented in section 5.2.1. Figure 4.3.3 shows two examples of projective vector sum surfaces constructed by two IFS-curves. Point weight editing has been applied on the control points of the IFS-curves defining the surface.

Below, figure 4.3.4 shows an advanced vector sum surface design, which illustrates the design capabilities of this method. More iterative surface design examples will be discussed within the scope of the following chapter (cf. section 5.4).

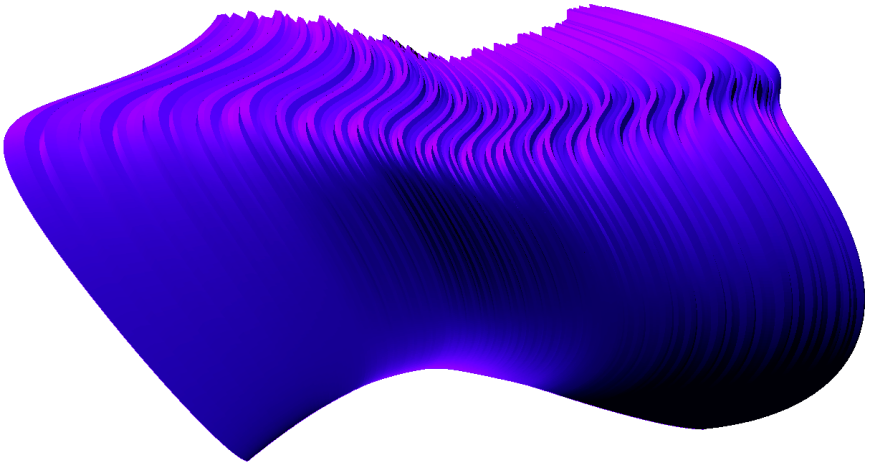


FIGURE 4.3.4. Example of a Vector sum Surface

The presented surface method offers high design potential. It unifies, in one formalism, the hitherto separate paradigms of the “smooth” and the “rough”. Furthermore, it verifies a certain number of geometric constraints, allowing the optimization of the production of free-form architecture.

Geometric Design by Discrete Iterative Geometry

In section 4.1, transformation driven geometric design has been explained by the example of an iteratively constructed curve, which has three control points and two transformations. However, the developed design formalism is far more general and allows designing a large variety of different geometric figures. Initially, we intended to create a geometric design method allowing to design within one formalism subdivision curves and surfaces, L-systems, fractals and classical smooth objects such as Bézier- or NURBS-curves. In order to show that the developed method of iterative geometric design allows the generation of the aforementioned objects, several geometric design examples using discrete iterative geometry will be presented within the following. Hereby, specific notions of iterative geometric design are going to be addressed such as resolution control, multi-patch modeling, edge-orientation, point-weight editing and subdivision point editing.

5.1. Topological Aspects

Within this section, discrete iterative figures are composed of a set of lines. Generally, iteratively designed figures do not provide continuous objects, meaning that the elements composing the figures are not constrained to be connected to each other. We will discuss the case of disconnected figures by the example of the Cantor set. By a set of variations and the manipulations of the subdivision points we will show how iterative geometric design can actually be used for the design of continuous curves. Hereby, we discuss how topological aspects of the obtained figures may be integrated into the subdivision scheme of iterative geometric design.

5.1.1. Variations on the cantor set. Now that the principles of transformation driven geometric design have been presented, parametrized Cantor sets can be created. Figure 5.1.1 shows the IFS-object of a parametrized

Cantor set being controlled by two control points $\{p_1, p_2\}$ and four subdivision points $\{s_1, s_2, s_3, s_4\}$. Therefore, the transformations $\{T_1, T_2\}$ are defined by two 2x2 matrices:

$$T_1 = (c_1, c_2) = \begin{pmatrix} c_{1,1} & c_{2,1} \\ c_{1,2} & c_{2,2} \end{pmatrix}$$

$$T_2 = (c_3, c_4) = \begin{pmatrix} c_{3,1} & c_{4,1} \\ c_{3,2} & c_{4,2} \end{pmatrix}$$

The free parameters of the transformation matrices are the two columns $\{c_2, c_3\}$. By modifying the subdivision points $\{s_2, s_3\}$, the scaling ratio of the line segments from one construction step to the other will be affected. The IFS-object defined herewith produces always Cantor sets, but not necessarily preserving the 1/3, 1/3, 1/3 ratio.

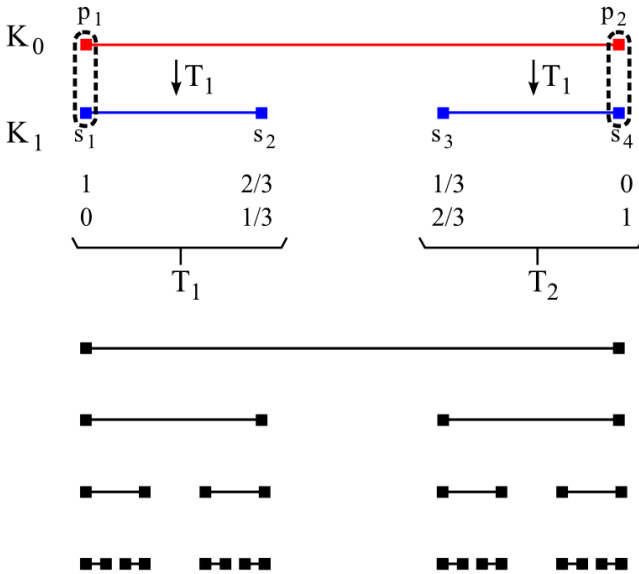


FIGURE 5.1.1. Cantor Set

If the two subdivision points $\{s_2, s_3\}$ are coinciding, the iterated result will provide a continuous straight polygonal line segment reaching from p_1 to p_2 .

Until now the actual IFS-object is restricted and will not allow other than changing the scaling ratio of the subdivided line segments. The example shown in figure 5.1.2 presents an un-constrained version of the

cantor set. All the values of the transformation matrices can be modified by the manipulation of the four subdivision points $\{s_1, s_2, s_3, s_4\}$. This is providing some more design possibilities such that the resulting version of the cantor set is not constrained to include the control points.

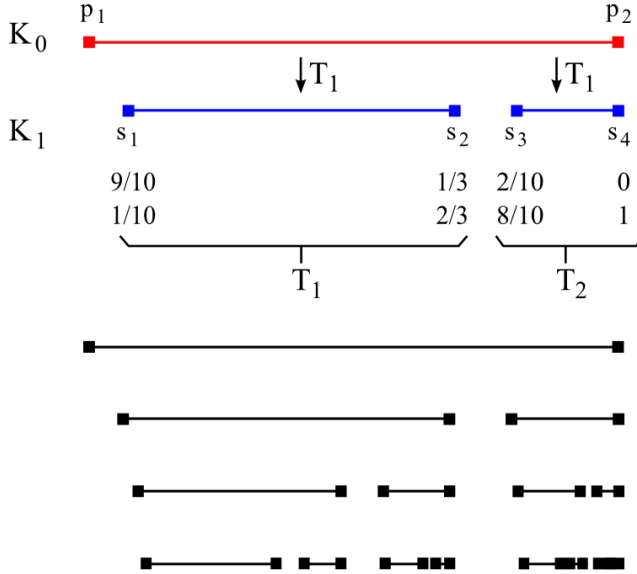


FIGURE 5.1.2. Un-constrained Cantor Set

The two examples mentioned above will always create a set of line segments that are elements of the line defined by the two control points $\{p_1, p_2\}$. The generated elements will never exceed the space defined by $\{p_1, p_2\}$ being a one dimensional straight line. The points $\{p_1, p_2\}$ present the barycentric coordinate system. In order to obtain variations of more-dimensional cantor sets, the number of control points needs to be increased.

Figure 5.1.3 illustrates a two-dimensional cantor-set. The use of three control points $\{p_1, p_2, p_3\}$ allows the design of figures in the 2D-plane, being defined by the three control points.

Since we work with three control points, 3×3 transformation matrices $\{T_1, T_2\}$ are used. In the present example, constraints have been put on the two subdivision points $\{s_1, s_6\}$ such that they equal the control points $\{p_1, p_3\}$. Therefore, the transformation matrices present only four free columns, such that:

$$T_1 = (c_1, c_2, c_3) = \begin{pmatrix} 1 & c_{2,1} & c_{3,1} \\ 0 & c_{2,2} & c_{3,2} \\ 0 & c_{2,3} & c_{3,3} \end{pmatrix}$$

$$T_2 = (c_4, c_5, c_6) = \begin{pmatrix} c_{4,1} & c_{5,1} & 0 \\ c_{4,2} & c_{5,2} & 0 \\ c_{4,3} & c_{5,3} & 1 \end{pmatrix}$$

The subdivision scheme presented below, shows fixed values for the transformation matrices $\{T_1, T_2\}$. They are conducting to the example a) of figure 5.1.4. The more general expression of the subdivision scheme (where the parameters $\{c_i\}$ are defined by the subdivision points) is given by the transformations matrices above.

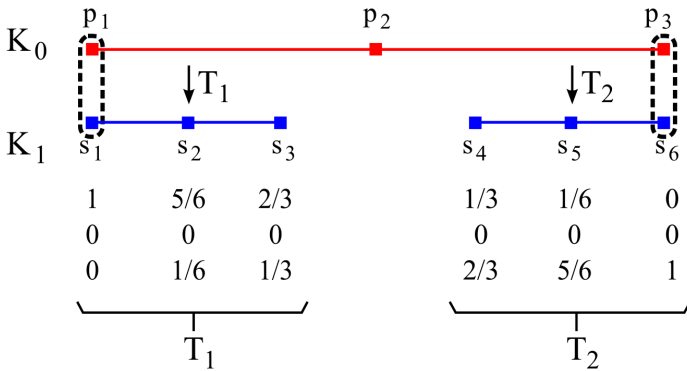


FIGURE 5.1.3. Two-dimensional Cantor Set

The scheme of the present IFS allows the generation of 2D cantor set variations. Figure 5.1.4 shows a selection of such variations.

Example a) of figure 5.1.4 presents the original cantor set. If, and only if, the two subdivision points $\{s_3, s_5\}$ are element of the line defined by $\{p_1, p_3\}$, the design possibilities equal those of the 1D cantor set presented by figure 5.1.1. More generally, the 2D cantor set allows to manipulate the subdivision points within the plane $\{p_1, p_2, p_3\}$ such that it is possible to design curved cantor sets such as shown in example b). Furthermore, particular configurations of the subdivision points allows the design of plane cantor sets as shown by case c) of figure 5.1.4).

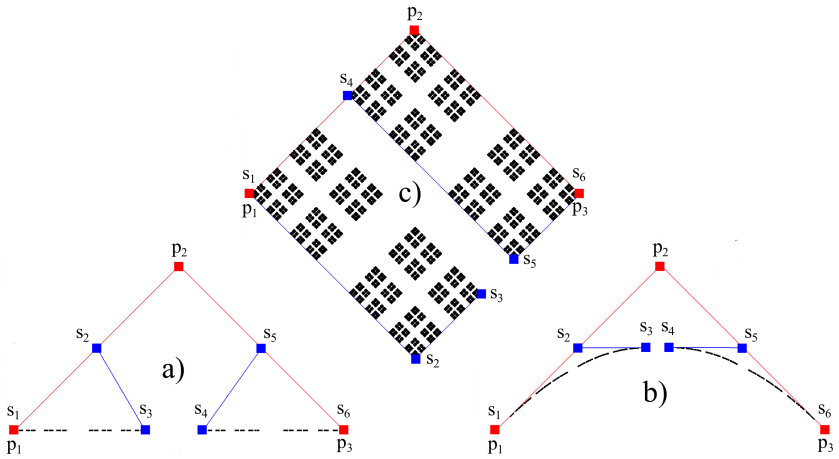


FIGURE 5.1.4. Cantor set variations

5.1.2. Connected figures. The discussed IFS-object defined by figure 5.1.3 is generally producing cantor-like point figures. Since there are absolutely no constraints on the free values of the subdivision matrices, the subdivision points can be freely positioned in the model space. In some particular cases, it is possible to leave the domain of the cantor set (disconnected figures) and to design other objects such as continuous curves or fractal self-intersecting figures.

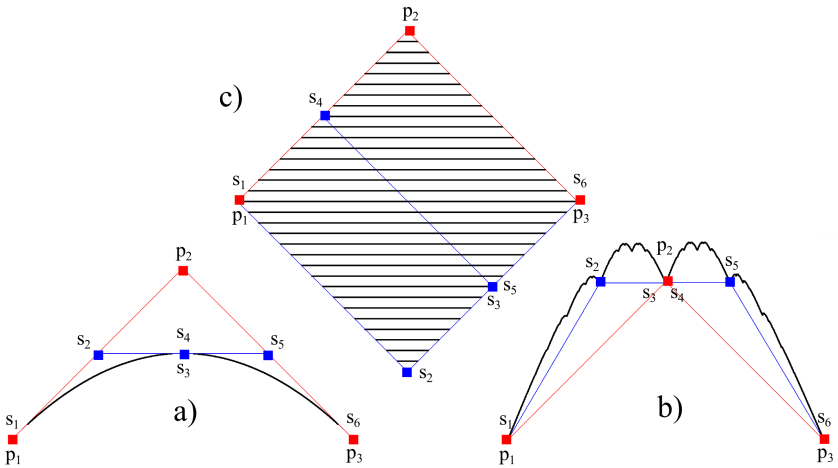


FIGURE 5.1.5. Coinciding subdivision points: Continuous IFS-curves a) and b), plane filling IFS-object c)

Coinciding subdivision points (cf. figure 5.1.5): If, and only if the subdivision points $\{s_3, s_5\}$ are coinciding, the resulting figure will be a continuous curve, as e.g. the Bézier-curve (shown in case a) of figure 5.1.5) or the Takagi-curve (shown in case b) of figure 5.1.5). Even though the IFS-object does not guarantee the produced figure being continuous, it is possible to assign the same coordinates to two different subdivision points.

Example c) of figure 5.1.5 shows us s_3 coinciding with s_5 . Under that condition, in some particular situations, plane filling figures may be designed. Case c) of figure 5.1.5 would lead to a plane filling object at its limit state A. For the representation, the discrete figure has been represented at the iteration level 9.

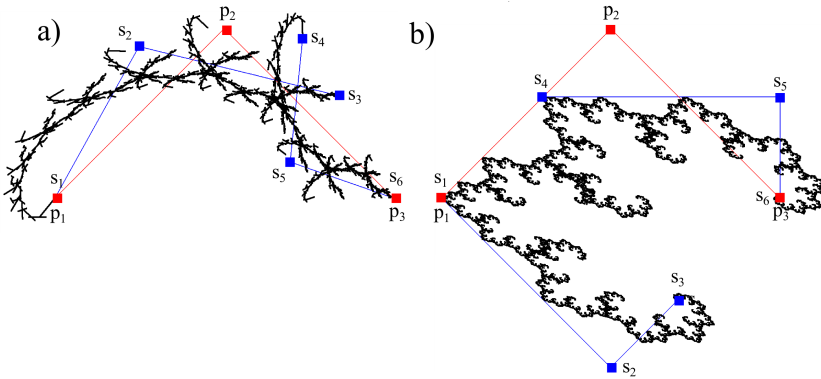


FIGURE 5.1.6. Self-intersecting fractal objects

Finally, two fractal figures shall be presented: Case a) and b) of figure 5.1.6. In this example, none of the subdivision points are coinciding. Despite of that fact, the produced figures are not of the domain of the cantor set but present self intersecting fractal figures. These two examples are interesting because they illustrate the large variety of shapes that can be designed by this IFS-object.

5.1.3. Notes. We have seen by the example of the cantor set how disconnected figures can be designed by iterative geometry. The use of more than two control points allows to quit the domain of the straight line and to begin the design of figures in the plane. In consequence, the use of four or more control points would actually allow the design of spatial figures including completely space filling figures. The link between point figures and curves (which will be the subject of the next section) has

shown to be two coinciding subdivision points $\{s_3, s_4\}$, what has been addressed as spacial cases under 5.1.2. Therefore, curves can actually be considered as topologically restricted IFS figures.

5.2. Curve Design

After having discussed some particular IFSs creating disconnected objects, a couple of geometric design methods using discrete iterative geometry for the design of curves is going to be presented. As stated above, curves are designed by constrained IFSs. Constraints on the subdivision point must verify that the created figure will be continuous. By constraining at least two subdivision points, i.e. $s_3 = s_4$, this property is verified (cf. figure 5.2.1).

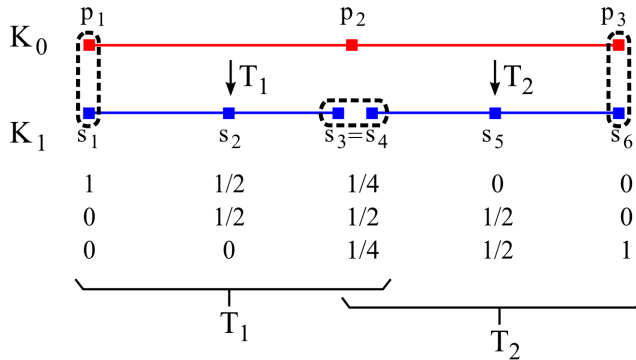


FIGURE 5.2.1. Continuous IFS-curve

Accordingly, the general scheme of the transformation matrices $\{T_1, T_2\}$ is:

$$T_1 = \begin{pmatrix} 1 & c_{2,1} & c_{3,1} \\ 0 & c_{2,2} & c_{3,2} \\ 0 & c_{2,3} & c_{3,3} \end{pmatrix}, T_2 = \begin{pmatrix} c_{3,1} & c_{4,1} & 0 \\ c_{3,2} & c_{4,2} & 0 \\ c_{3,3} & c_{4,3} & 1 \end{pmatrix}$$

Based on this subdivision scheme, following curve objects will be discussed:

- First, Bézier curves will be explained. As known, quadratic Bézier curves can only be used to design parabolic segments.
- Second, the use of rational Bézier curve will allow creating conics, that are used for the design of circles, ellipses and, more generally, circular arcs.

- Third, subdivision point editing will be discussed in order to show the design method for the construction of some basic fractal objects, such as, the Lévy-C curve, the von Koch curve, the Takagi curve and finally an interpolating Bézier curve.

These examples are particular curves of a broader curve family, that can be designed using the same IFS-object. Later, design methods for creating higher order curves such a cubic Bézier curves and cubic B-Splines will be considered. Finally, more special figures, like the Dragon curve, are discussed in order to introduce to the problem of line segment orientation within the designed figure.

5.2.1. Quadratic Bézier Curves. The example of a continuous curve constructed by two transformations as described by figure 5.2.1 will lead to a discrete representation of a quadratic Bézier curve, representing a parabolic segment (cf. top row of figure 5.2.2). The Bézier curve can be manipulated using the traditional design aids, namely the control points P , which can be freely manipulated in the design space (usually \mathbb{R}^3). In addition to this, point weight editing will allow the design of rational Bézier curves¹ (cf. bottom rows of figure 5.2.1).

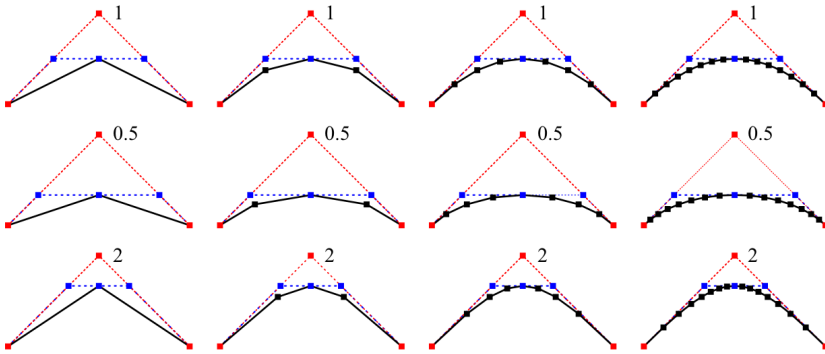


FIGURE 5.2.2. Rational quadratic Bézier curves: Iterative construction steps $n = \{1, 2, 3, 4\}$; influence of point weight editing

Note that we always obtain a discrete expression of a Bézier curve. The refinement of the obtained figure can be controlled by the level of

¹The notion of point weight editing refers to the use of homogeneous coordinates (w, x, y, z) . Please consider section 4.3.2, where the relation of homogeneous \mathbb{R}^4 -coordinates and Cartesian \mathbb{R}^3 -coordinates is presented.

iteration. Generally, after 5 iterative construction steps, we end up with an object that will be perceived as sufficiently smooth.

5.2.2. Conics by Rational quadratic Bézier Curves. Rational Bézier curves allow the edition of the control points weight at the designers desire, which is important for the design of certain particular figures. Non rational cubic Bézier curves are only able of representing parabolic arc segments. If one would like to design specific geometric objects, such as circles for instance, non-rational cubic Béziers will not suffice. Based on the findings of Blanc and Schlick [BS96] there exist accurate methods for the representation of conics using rational Bézier curves. With this knowledge, we are able to design not only free-form Bézier curve patches, but also a large variety of specific geometrical figures such as circles or ellipses and, more generally speaking, conics. The design of conics constrains the control points to specific values with regards to their position in space and their respective control point weight as shown in figure 5.2.3.

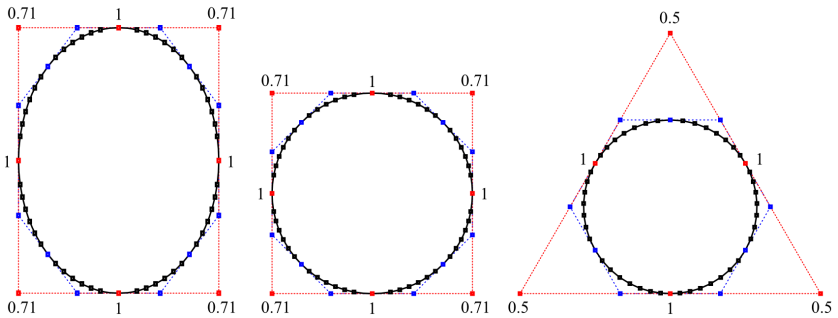


FIGURE 5.2.3. Representation of conic curves using multiple patches of discrete IFS-curves. The numbers shown in the figure correspond to the w -values of the respective control points.

The examples showed in figure 5.2.3 use several rational quadratic Bézier curves that are connected together in order to create circles or ellipses. Figures that are composed of several sub-curves are called multi-patch figures, where one sub-curve is regarded as one patch. When working with multiple patches, the control polygons P of the patches have generally one or several control points p_i in common with its neighboring patches. More examples of multi-path figures are addressed in section 5.4.2 presenting the case of multi-patch surfaces.

Drawing figures like circles or ellipses can be basically brought down to the problem of designing a series of discrete circular arcs, more generally, conical segments.

Nota bene: The circle representations shown in figure 5.2.3 are not the only ones employed for the design of circles using multiple Bézier patches. Though, as far as known, they are the only ones using quadratic Bézier curves exclusively.

5.2.3. From quadratic Béziers to fractal Curves. Up to now, we have discussed some design methods of classic rational Bézier curves. The figures are entirely defined by the position of its control points in the design space and the values of the point weights assigned to them. In order to get from smooth curve figure to fractal figures, subdivision point editing will be employed.

We consider the manipulation of figures via the control points P as global shape control. For example, control point editing allows the manipulation of the curve's end and start points, which is providing control over how the figure will be curved globally. Point weight editing allows acting more locally on the figure assigning more or less "importance" to the individual control points. However, the local aspect of the curve designed remains unchanged.

The control of the local aspect of the figure can be achieved by the manipulation of the subdivision points S . Subdivision points provide control of the figure's smoothness and roughness respectively. Based on the example of a continuous curve described by figure 5.2.1, this paragraph will show a series of fractal curve design using subdivision point editing.

In order to limit the number of free parameters of the transformations, we propose to merge some of the subdivision points s_i with some of the control points p_i . Working with three control points $\{p_1, p_2, p_3\}$ and six subdivision points $\{s_1, s_2, s_3, s_4, s_5, s_6\}$, we will merge the following points such that following three relations are verified:

$$p_1 = s_1, p_2 = s_3 = s_4, p_3 = s_6$$

Herewith, the scheme of the transformation matrices $\{T_1, T_2\}$ is:

$$T_1 = (c_1, c_2, c_3) = \begin{pmatrix} 1 & c_{2,1} & 0 \\ 0 & c_{2,2} & 1 \\ 0 & c_{2,3} & 0 \end{pmatrix}, T_2 = (c_4, c_5, c_6) = \begin{pmatrix} 0 & c_{5,1} & 0 \\ 1 & c_{5,2} & 0 \\ 0 & c_{5,3} & 1 \end{pmatrix}$$

The columns c_2 and c_5 are the only free parameters which can be defined by the position of the subdivision points s_2 and s_5 in the design

space. Remember: The free columns c_i of the transformation matrices can be computed by

$$c_i = P^{-1}s_i$$

Figure 5.2.4 shows the iterative construction of four IFS curves. The position of the control points and the subdivision points $\{p_i, s_i\}$ used for the construction of the curves are shown on the left column of the figure. Note that only the subdivision points are manipulated, all the control points and the point weights remain strictly identical for all of the four examples.

At construction step 1, shown in the second column from the left, the resulting discrete curves are all exactly the same. K_1 matches in all the present cases the control point polygon. At construction step 2, shown in the third column from the left, the resulting figures match the subdivision point polygon. At this stage, the curves are all different objects, geometrically speaking. The right column shows the discrete curve at construction step ten, providing already an accurate image of the limit state A towards which the figures are converging.

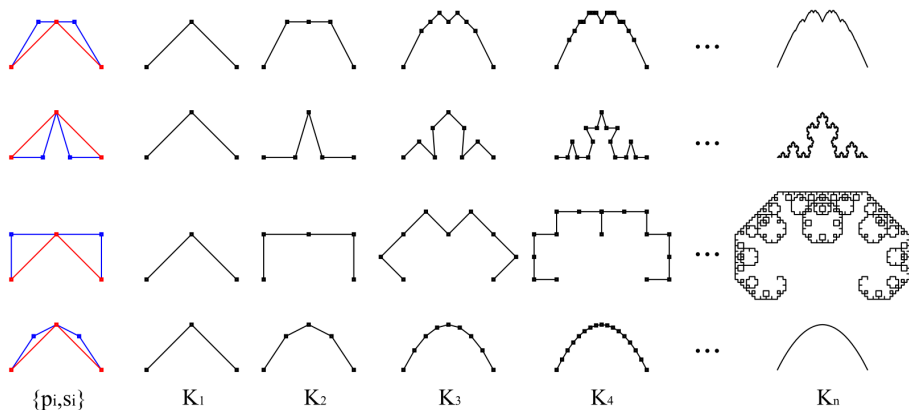


FIGURE 5.2.4. Subdivision point editing

The four chosen examples are all famous curves. In the top row, the Takagi curve is shown, which Takagi reported for the first time in [Tak03]. This curve is also named blancmange curve. Blancmange is a brand of instant pudding. The shape of the curve resembles actually the shape of the pudding presented on an advertising poster of the blancmange pudding. The second row of figure 5.2.4 shows the construction of a von Koch curve, which has already been presented in section 3.2. Row three shows

the Lévy C-curve, on which one can find a English transcript in [Lév93] of Lévy's french article that was published in 1938. Finally, the bottom row shows a parabolic segment that is similar to the ones that could be designed by quadratic Bézier curves discussed in paragraph 5.2.1. The difference is that the parabolic segment shown in figure 5.2.4 is an interpolating curve, which means that it passes through the three control points $\{p_1, p_2, p_3\}$.

Although the four curves are generally treated as separate distinct figures, iterative geometric design treats the shown four examples as one IFS object. The parameters of the IFS-object differ from one curve to the other because of the changing position of the subdivision points.

The discrete representation of smooth figures such as the aforementioned parabolic segment demands less iterations in order to obtain an accurate rendering of the curve. Generally, after four or five construction steps, the designer will not recognize any significant changes of the figures shape. Fractal and folded figures, like i.e the Lévy C-curve, need much more construction steps. Single line segments of the Lévy curve can still be distinguished after ten iterations (column K_n of figure 5.2.4), while the shape of the Bézier curve K_4 does not evolve significantly compared to K_n .

Figure 5.2.5 shows a more more exhaustive table of variations of the same IFS-object. It is interesting to state that the present IFS-object includes the special cases of the Von Koch curve, the Lévy-C curve, the Takagi curve and a plane filling curve. They present limit cases which may be obtained by precise positioning of the subdivision points

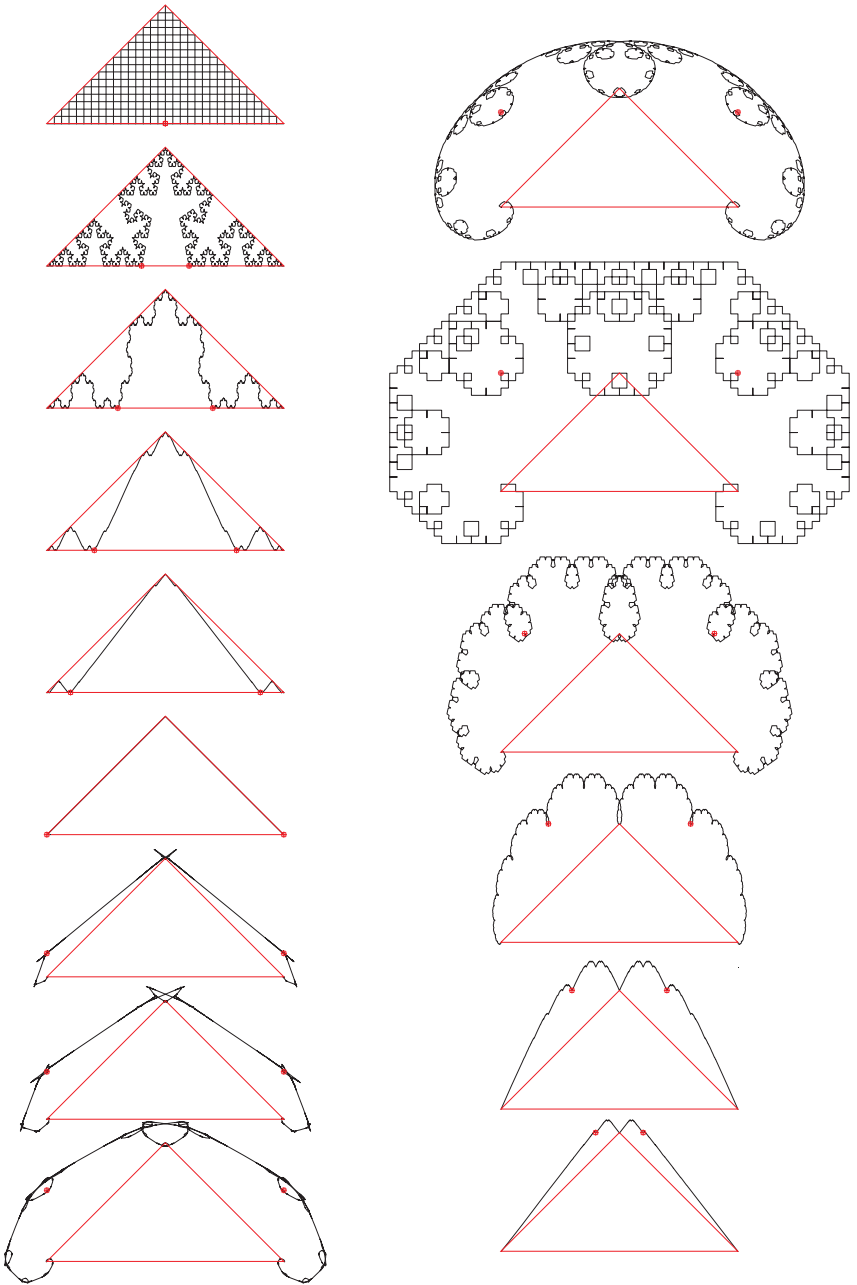


FIGURE 5.2.5. Subdivision point modeling providing variations of an IFS-curve

5.2.4. Cubic Bézier curves and Cubic B-Splines. This section shows how higher order curves like for example cubic Bézier curves or cubic B-Splines can be constructed iteratively. Both mentioned objects are continuous curves built by two transformations $\{T_1, T_2\}$.

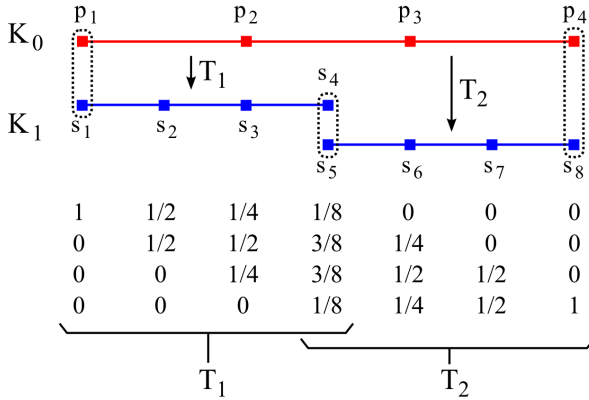


FIGURE 5.2.6. Cubic Bézier curve

Classically, Bezier curves are differentiated by the degree, which actually does not make much sense in terms of iterative geometric design because no explicit algebraic expression is used for the design of iterative curves. However, the similarity, that might be established, concerns the number of control points used for a given geometric curve design.

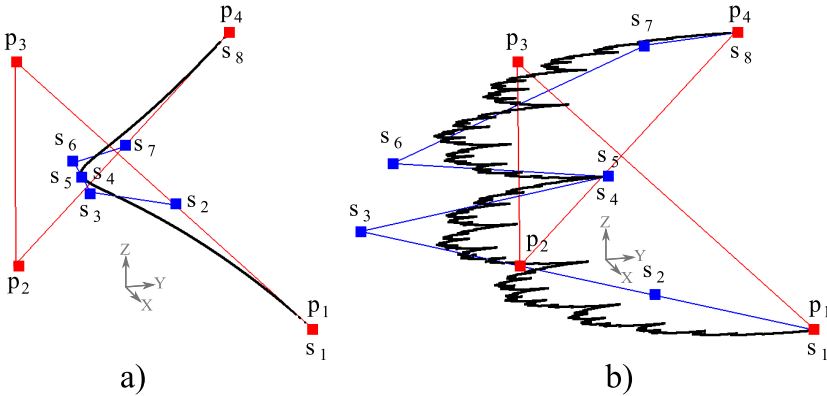


FIGURE 5.2.7. a) Iteratively designed cubic Bézier curve; b) Cubic Takagi curve

Contrary to the previously discussed figures, the two examples discussed within the scope of this section are controlled by four control points $\{p_1, p_2, p_3, p_4\}$. The use of four control points allow to quit the domain of the 2D plane² and to begin the design of spacial 3D-curves. Figure 5.2.7 shows two curves designed by the IFS-object described in 5.2.6. While case a) illustrates a classical discrete cubic Bézier curve, case b) illustrates a three dimensional form of a Takagi curve. Both resulting figures rely upon the same IFS-Object. Only their geometric values of the subdivision points differ.

One important difference between a cubic Bézier curve and a cubic B-Spline lies in the way in which the two transformed patches $\{K_0T_1, K_0T_2\}$ are connected, cf. figures 5.2.6 and 5.2.8. While the two transformed control polygons have only one point in common in the case of the Bézier curve $\{s_4 = s_5\}$, they have three points in common in the case of the B-Spline $\{s_2 = s_5, s_3 = s_6, s_4 = s_7\}$.

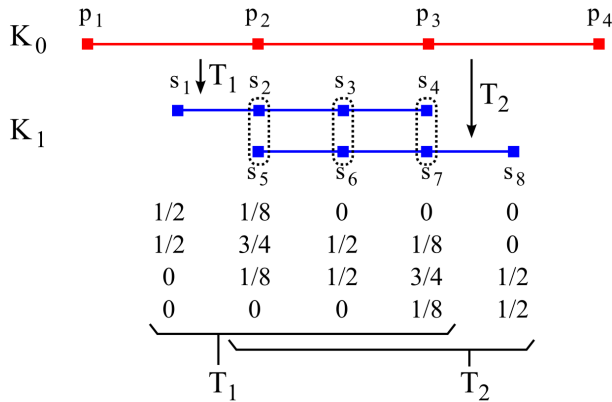


FIGURE 5.2.8. Cubic B-Spline

This topological constraint has certain consequences on the generated figures, as it implies a higher level of continuity. Generally, curves with high level of continuity are smoother than simply connected figures. Practically, this means that iterative geometric figures of high level of continuity are more difficult to “fractalize” than C0 figures, which have just one single subdivision point merged.

²to which figures defined by three control points $\{p_1, p_2, p_3\}$ are restricted

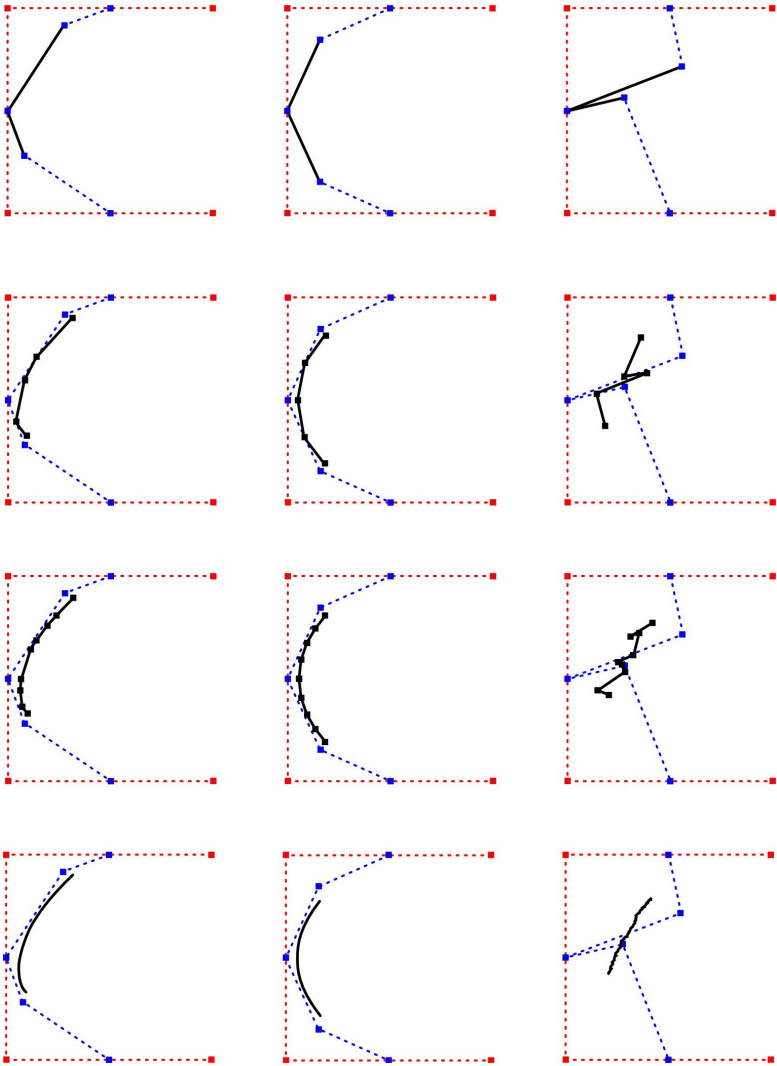


FIGURE 5.2.9. Subdivision point editing of a cubic B-Spline curve shown in the middle column

Figure 5.2.9 shows a cubic B-Spline in the middle column. The effect of the subdivision point editing is rather subtle on the example shown in the left column. Even by important deformations of the subdivision polygon, shown on the right, the produced figure at construction step 4 is far less fractal like than the fractal curves shown in section 5.2.3.

5.3. Orientation and Symmetry

For the iterative geometric design, the order of the subdivision points is of high importance for the resulting figure. In some particular situations, the designed figures do not show an apparent difference with regards to the order in which the subdivision points have been introduced. In other cases the result obtained will differ completely depending the subdivision points' order.

Within the following, two pairs of examples have been selected in order to discuss the problem of the subdivision points' order. The configuration of the control points p_i and the subdivision points s_i used for these examples is graphically illustrated by figure 5.3.1.

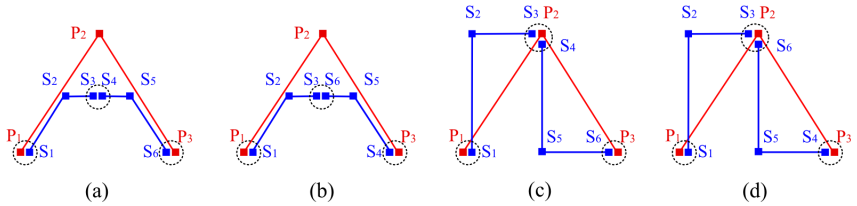


FIGURE 5.3.1. (a) $p_1 = s_1, s_3 = s_4, s_6 = p_3$ (b) $p_1 = s_1, s_3 = s_6, s_4 = p_3$ (c) $p_1 = s_1, p_2 = s_3 = s_4, s_6 = p_3$ (d) $p_1 = s_1, p_2 = s_3 = s_6, s_4 = p_3$

As we can see, only the order of the subdivision points $\{s_4, s_5, s_6\}$ has been changed in these examples. Below are listed the constraints that will condition the transformation matrices:

- a) $p_1 = s_1, s_3 = s_4, s_6 = p_3$
- b) $p_1 = s_1, s_3 = s_6, s_4 = p_3$
- c) $p_1 = s_1, p_2 = s_3 = s_4, s_6 = p_3$
- d) $p_1 = s_1, p_2 = s_3 = s_6, s_4 = p_3$

By the law $c_i = P^{-1}s_i$ and the conditions listed above, the schemes of transformation matrices are:

$$\text{a) } T_1^a = (c_1, c_2, c_3) = \begin{pmatrix} 1 & c_{2,1} & c_{3,1} \\ 0 & c_{2,2} & c_{3,2} \\ 0 & c_{2,3} & c_{3,3} \end{pmatrix}, T_2^a = (c_3, c_5, c_6) = \begin{pmatrix} c_{3,1} & c_{5,1} & 0 \\ c_{3,2} & c_{5,2} & 0 \\ c_{3,3} & c_{5,3} & 1 \end{pmatrix}$$

$$\begin{aligned}
 \text{b) } T_1^b &= (c_1, c_2, c_3) = \begin{pmatrix} 1 & c_{2,1} & c_{3,1} \\ 0 & c_{2,2} & c_{3,2} \\ 0 & c_{2,3} & c_{3,3} \end{pmatrix}, T_2^b = (c_6, c_5, c_3) = \begin{pmatrix} 0 & c_{5,1} & c_{3,1} \\ 0 & c_{5,2} & c_{3,2} \\ 1 & c_{5,3} & c_{3,3} \end{pmatrix} \\
 \text{c) } T_1^c &= (c_1, c_2, c_3) = \begin{pmatrix} 1 & c_{2,1} & 0 \\ 0 & c_{2,2} & 1 \\ 0 & c_{2,3} & 0 \end{pmatrix}, T_2^c = (c_3, c_5, c_6) = \begin{pmatrix} 0 & c_{5,1} & 0 \\ 1 & c_{5,2} & 0 \\ 0 & c_{5,3} & 1 \end{pmatrix} \\
 \text{d) } T_1^d &= (c_1, c_2, c_3) = \begin{pmatrix} 1 & c_{2,1} & 0 \\ 0 & c_{2,2} & 1 \\ 0 & c_{2,3} & 0 \end{pmatrix}, T_2^d = (c_6, c_5, c_3) = \begin{pmatrix} 0 & c_{5,1} & 0 \\ 0 & c_{5,2} & 1 \\ 1 & c_{5,3} & 0 \end{pmatrix}
 \end{aligned}$$

Since only the subdivision points $\{s_4, s_5, s_6\}$ have been changed, only the second transformation matrix T_2 is affected by the changed subdivision point order. By swapping s_4 and s_6 the values of the first and the last columns of T_2 will swap accordingly. In other terms, the orientation of the subdivision point polygon is modified. The representation of the subdivision points' order could also be shown by the orientation of the line segments forming the control polygon. To do so, we order the pair of vertices which define a line segment of the control polygon. In figure 5.3.2, the orientations of the line segments are represented by arrows.

The examples a) and b) show once more the iterative construction of a quadratic Bézier curve. Looking at the resulting figures K_n , they seem to be identical. This might be true from a purely geometrical point of view: The coordinates of every vertex of figure a) correspond to the ones of a certain vertex of figure b). However, the order of the line segments building up the figure has radically changed. For now we will say that the two figures a) and b) are geometrically the same. The influence of the changed order will appear e.g. in terms of addressing, which will be discussed later.

Example d) shows the dragon curve, which is a very famous curve that has been discussed by many researchers in the field of fractal geometry [Bar88, Fal90, VG95]. The reason why we use the dragon curve in this context is because this curve can only be constructed with precisely oriented subdivision control points. The effect of the subdivision point's order can be illustrated in comparison to the example curve c), which used exactly the same coordinates for its subdivision points but their order is different (cf. figure 5.3.1). With regards to the resulting figure, the effect of the changed order shows up from construction state K_3 and onwards. If the orientation of the line segments was considered, the difference would already be present at earlier construction states.

However, the figures produced after ten construction steps ($n = 10$) illustrate well the matter of the orientation of the subdivision points, since

they lead in the case of the two examples c) and d) to geometrically completely different resulting figures.

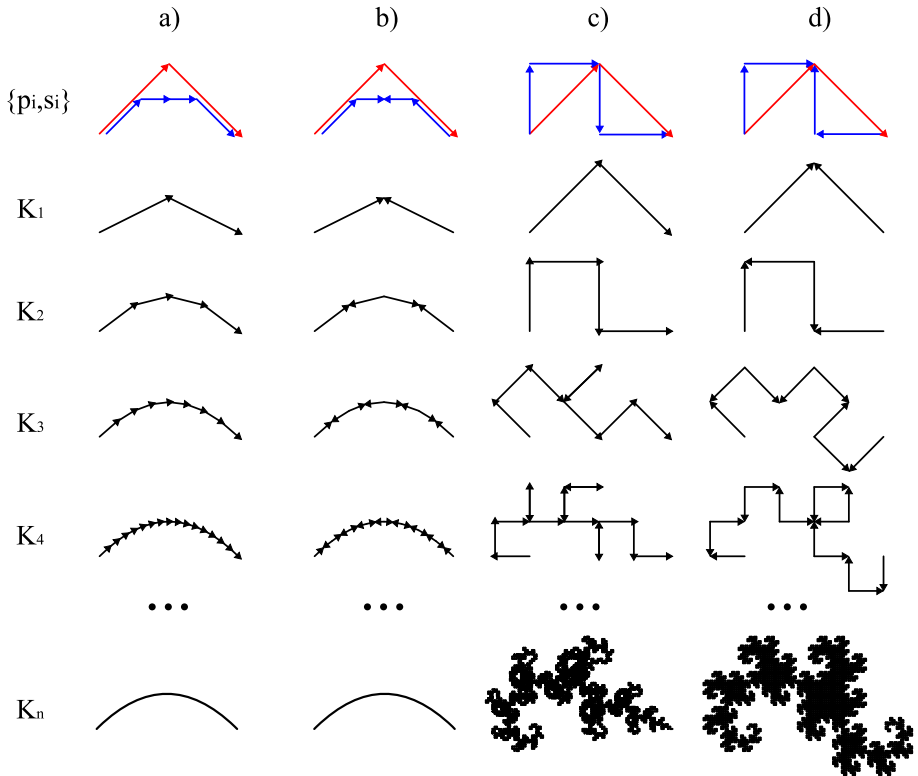


FIGURE 5.3.2. The orientation of the line segments influences the resulting figure

The symmetrical layout of the subdivision points $\{s_1, s_2, s_3\}$ and $\{s_6, s_5, s_4\}$ in the examples $\{a, b\}$ lead geometrically identical curves. The asymmetrical layout of the same subdivision point in the examples $\{c, d\}$ lead to geometrically different figures. Therefore, the asymmetry combined with inverted order of $\{s_i\}$ is affecting the geometry of the resulting figure. Inverting the order combined with a symmetrical layout of s_i affects only the order of the elements composing the figure designed.

5.4. Surfaces and Operators

In this section we will provide some surface methods to illustrate how IFS-surfaces can be designed and manipulated. Basically, the presented surfaces are produced by the combination of two curves.

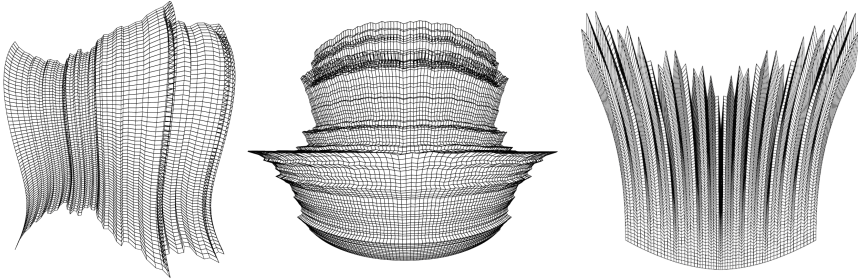


FIGURE 5.4.1. Examples of iteratively designed surfaces

5.4.1. Surfaces by curve products. The surfaces shown in figure 5.4.2 are the product of two curves $\{A, B\}$. The employed curves are the same as the ones presented in 5.2.3, which are controlled by three control points $\{P^A, P^B\}$, six subdivision points $\{S^A, S^B\}$ and two transformations $\{T_1^A, T_2^A\}$ and $\{T_1^B, T_2^B\}$. The operator creating sub-dividable surfaces is a tensor product which uses the curves parameters to obtain the surface parameters. This method has been presented by Prautzsch and Michelli in [MP87].

The surface is controlled by a grid of nine control points $\{P\}$ and 36 subdivision points $\{S\}$ and four transformations $\{T_1, T_2, T_3, T_4\}$. The product of the curves' transformation matrices $\{T_1^A, T_2^A\}$ and $\{T_1^B, T_2^B\}$ provide four 9×9 matrices $\{T_1, T_2, T_3, T_4\}$, which define the way in which the surface will be subdivided:

$$\{T_1 = T_1^A \otimes T_1^B, T_2 = T_2^A \otimes T_1^B, T_3 = T_1^A \otimes T_2^B, T_4 = T_2^A \otimes T_2^B\}$$

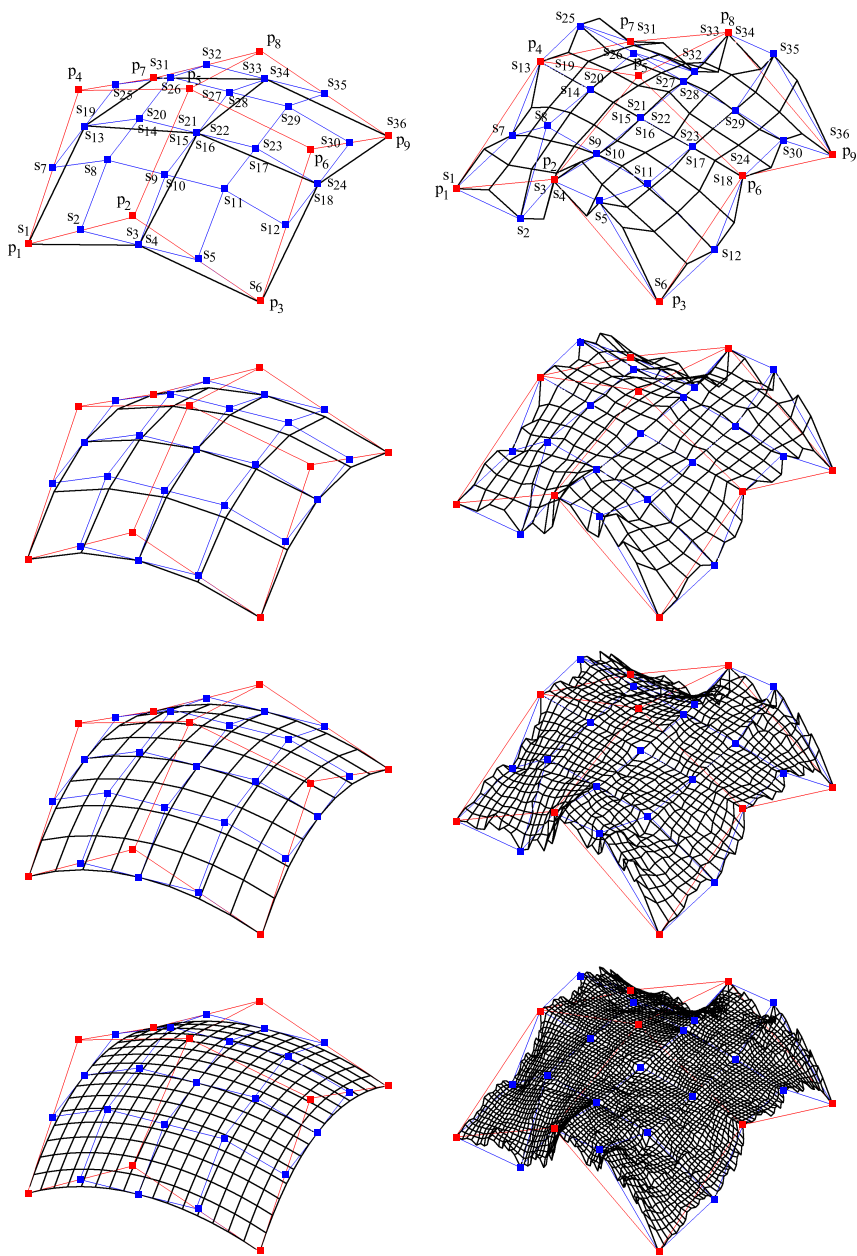


FIGURE 5.4.2. 3x3 tensor product surface: Bézier-patch on the left, fractal-patch on the right

Once the subdivision matrices are defined they can be applied on the control grid $\{P\}$ in order to compute the subdivision points $\{S\}$. The input curves $\{A, B\}$ are not used anymore. The design of the resulting IFS-surface can henceforth be achieved by the manipulation of the control and subdivision grids $\{P, S\}$, which will act on the previously defined transformation matrices $\{T_1, T_2, T_3, T_4\}$.

Using this method, it is obviously possible to design classical Bézier patches (shown in the left column of figure 5.4.2), which are defined by the curve product of two quadratic Bézier curves. The Bézier patch is a particular case of the presented IFS-surface method. By the manipulation of the subdivision points, fractal and rough surfaces will generally be obtained; shown in the right column of figure 5.4.2.

In addition to the control points shown in red, iterative surface design provides a subdivision control grid, shown in blue. The sum of the two point grids $\{P, S\}$ results in a heavily cluttered control point cloud, where the manipulation of the single points becomes difficult. This is actually the main disadvantage of the proposed surface method.

The present example shows a single 3×3 patch. While the method allows the creation of iterative surfaces by higher dimension patches, the number of control points increases accordingly. The possibility to work with multiple patches will further exacerbate the nature of the control point grids.

Examples of higher degree and multi-patch surfaces will be shown later in this section. An application of an IFS-surface designed by the product of two curves is presented in section 8.3.

The principle of surface design by the product of two curves is common to most CAD software. Vector sum surfaces - as presented in section 4.3 - can be described as particular tensor product surfaces. For vector sum surfaces, the control point grid P and the subdivision point grid S are given by:

$$p_{i,j} = c + \overrightarrow{cp_i^A} + \overrightarrow{cp_j^B}$$

$$s_{i,j} = c + \overrightarrow{cs_i^A} + \overrightarrow{cs_j^B}$$

where c is an arbitrary point element of the design space.

5.4.2. Multi-patch Surfaces by curve sums. Multi-patch surfaces are surfaces composed of several patches. In order to guarantee the continuity of the surface, the single patches of a multi-patch surface are designed using the same set of subdivision transformations. The examples

shown are based on vector sum surfaces, which have been explained before (cf. section 4.3).

Figure 5.4.3 shows several versions of multi-patch figures. The surface in the middle consists of two connected patches; the figure on the right of four patches. In the case of the four-patch surface, four curves $\{A, B, C, D\}$ define the surface. On the one side, we have the two curves $\{A, B\}$ represented by their control points $\{p_i^A, p_i^B\}$, while on the other side, the curves $\{C, D\}$ are represented by the control points $\{p_i^C, p_i^D\}$. For better understanding the patches of the surfaces shown by figure 5.4.3 are all identical. This allows to identify more easily the single patches composing the surface. The vector sum surfaces illustrating the purpose of multi-patch surfaces are computed in the model space \mathbb{R}^3 .

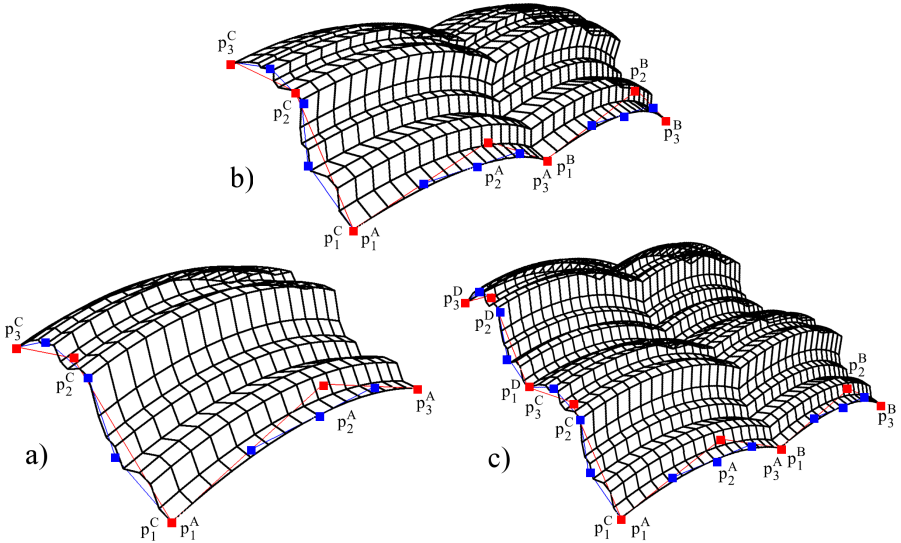


FIGURE 5.4.3. Multi-patch surfaces: a) One, b) two and c) four patches

The design possibilities offered by multi-patch surfaces are obviously far more diverse than simple copies of one patch, which was the case in the aforementioned example. Figure 5.4.4 shows the example of a cylindrical surface. On the left of the figure, the shape defining curves are rendered. The curve on the bottom is composed of three quadratic Bézier curve patches. They form a closed smooth curve. The profile curve is equally composed of three patches, constructing a rough and fractal profile curve. In this example the use of several patches was required for

the creation of a closed ground curve, which is impossible to create using a single quadratic Bézier curve patch. Further, projective geometry was employed to edit the point weight $\{w\}$ of the profile curve's control points. Hereby the vector sum surface is computed in the homogeneous space \mathbb{R}^4 and then projected into the model space \mathbb{R}^3 , which allows us i.e. to design cylindrical figures.

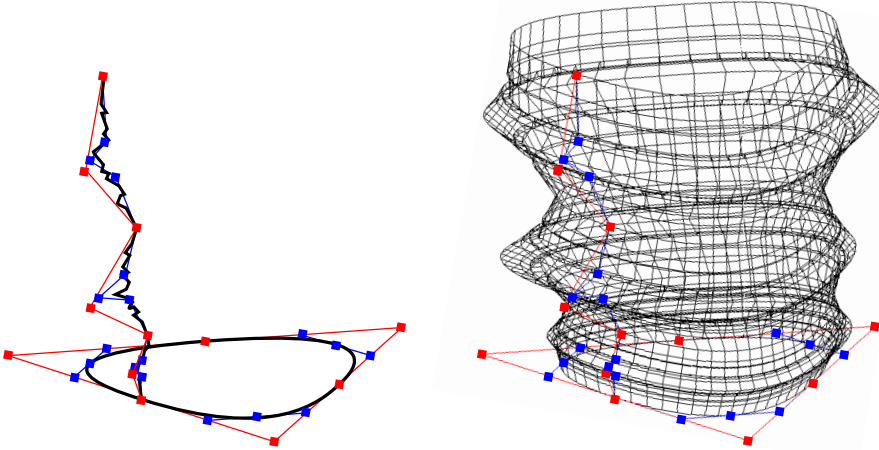


FIGURE 5.4.4. Fractal cylinder designed by multi-patch vector sum surfaces

Multi-patch geometric design is a common method to existing CAD-software.

5.4.3. Resolution control of vector sum surfaces. Based on the surface method presented in 4.3, the level of detail of the surface designed can be adjusted separately on each curve defining the figure. The example surface shown in figure 5.4.5 is controlled by two curves $\{A, B\}$, which are represented by their respective control points $\{p_i^A, p_i^B\}$. The example shows a IFS-surface, which has uneven proportions with regard to its width/length ratio. Generally, this produces a resulting quad mesh, which is composed of faces that are of rectangular proportions as opposed to being of square proportions.

For applications with regards to architectural design, it seems interesting to be able to change the level of iteration (= resolution of refinement) of each curve $\{i^A, i^B\}$ individually. E.g., if the surface was designed for a building's envelope, the size and shape of the cladding might need to be rather square than rectangular. The ability of defining different levels

of iteration for the curves allows controlling the proportion ratio of the elements composing the resulting surface.

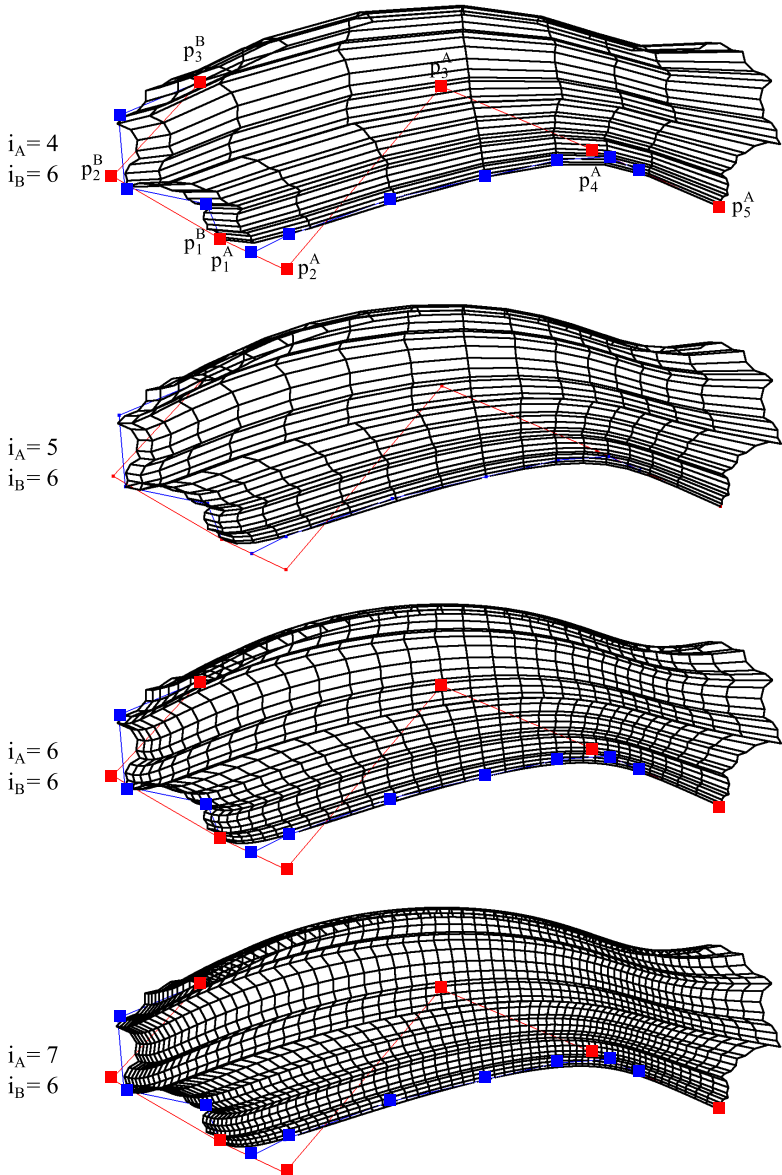


FIGURE 5.4.5. Vector sum surface by two curves of different levels of iteration

5.5. Discussion

The examples shown in this chapter illustrate a large family of figures, which may be designed by iterative geometric design. Besides classical fractal figures (like the cantor set, the von Koch curve or the Takagi curve), classical smooth figures (e.g. Bézier curves, conics or Splines) can equally be designed by the proposed geometric design formalism. In addition to classical curve objects, point figures, spatial curves or plane filling objects can be obtained.

The presented surface methods are based on vector sum surfaces and tensor product surfaces. Multi-patch modeling is illustrated as well as resolution control over the designed figure, allowing to work on a desired level of detail by increasing the refinement of a given figure. The presented geometric design methods can further be combined with each other and existing geometric design techniques may be used in combination with the addressed methods.

Part 3

Applications of Discrete Iterative Geometry to Architecture

This part will confront the iteratively constructed geometric figures with applications in the field of architecture. The following topics will be developed in order to use the mathematical method for architectural means:

- (1) The geometric figure will be translated into a set of constructional elements. This topic will be addressed in chapter 6.
- (2) The data describing the constructional elements will have to be conditioned with regards to the production of the elements. The production related questions are studied in chapter 7. The process, from geometric design to the fabrication of constructional elements, is often named “digital chain”, since it describes the work flow necessary to get from the geometrical description to the finished part.
- (3) The established work steps are tested on possible applications, which are presented in chapter 8. This chapter aims to illustrate the variety of design possibilities provided by the methods studied here.

In order to realize physical buildings out of discrete virtual geometries, the elements, which constitute the 3D-models, are replaced by constructional elements. For an iteratively designed curve, the line sections will be substituted by linear constructional elements, such as planks, girders or beams. In the case of a discrete surface, its faces are replaced by planar constructional elements (panels, plates, etc.). The substitution of geometric elements by constructional elements poses a certain number of questions as the geometric figures do not have physical dimensions like thickness. Chapter 6 studies the topic relevant to the conversion of the geometry data to construction data.

Once all the data describing the constructional parts has been calculated, the data set has to be conditioned for production. Several solutions exist, which allow partial automating of the production of any geometric object by means of integrated manufacturing. However, the production conditions such as material used, production process, employed machines and tools, remain important factors that influence greatly the set up of an efficient production process. In architecture, the final product is usually a prototype. Therefore, a particular building design and its constructional elements are built only once. Hence, optimization processes coming from the field of mass production, as used in the aeronautic and car industry, can therefore only be applied in a limited manner.

CHAPTER 6

From Geometry Data to Workshop Plans

6.1. Work Flow: Digital Chain

The concept of the digital chain relates to the process the design goes through in order to get from the initial design idea to the finished product. Furthermore, the digital chain refers to integrated design and manufacturing in order to optimize the product and shorten the links in the design and production cycles using computer technology. Our main concern is the structure of the design and production data, which are often of completely distinct nature.

In this chapter we are discussing how to condition the geometry data describing an architectural design in order to meet the requirements for the later production. Within the scope of this work, the digital chain organizes the data with regards to:

- Logistics - by means of addressing
- Geometry - in terms of shape and dimension describing the elements composing the product
- Topology - to differentiate the elements with regards to their function and relative position
- Production - defining a set of machining strategies

The last point will be treated within the scope of chapter 7. The established “digital chain” will have to allow design changes at any level of the product cycle.

6.2. Addressing

Why is addressing important? The topic of addressing the individual elements correctly is discussed in this section for two reasons: First, the geometrical elements of the figure need to be identified in order to apply the post treatment procedures necessary for computation of the data describing the constructional elements. Second, the element’s address will be necessary for later assembly of the parts on the construction site.

Generally, since all the elements building up the object are different in shape and size, the workers on site need a clear and comprehensive description of where to install which element. These two points, which briefly explain why we need to address every single element by unique identifiers, bring up a third question which is highly relevant for the design of the addressing system: the description of the neighborhood. For both means, a description of the context of each element is needed, which will define how many neighbors an element has and what the neighbors relative position is (e.g. upper, lower, left or right). Such a neighborhood description should be able to differentiate the elements with regards to their position in the global figure, whether the element is in the middle of the system or at its extremity.

The addressing needs therefore to meet following requirements:

- Bijective function, that relates one unique address to exactly one and only one element
- Order, providing neighborhood information.

The problem of addressing may be treated by the problem of data structuring, which is necessary to operate the methods allowing to automate the generation of constructional elements out of the geometry data provided by discrete iterative geometric design.

6.2.1. Addressing function provided by the IFS formalism. First, the generation of object identifiers by the geometrical design method employed for the construction of discrete IFS figures will be examined. The principles of the geometric design method have been discussed in chapter 4. For addressing means, we will discuss more in detail the construction sequence by the example of the continuous curve designed by two transformations $\{T_1, T_2\}$:

$$\begin{aligned}
 PK_0 &= PK \\
 PK_1 &= PT_1K_0 \cup PT_2K_0 = PT_1K \cup PT_2K \\
 PK_2 &= PT_1K_1 \cup PT_2K_1 = PT_1T_1K \cup PT_1T_2K \cup PT_2T_1K \cup PT_2T_2K
 \end{aligned}$$

Let us consider only the first two construction steps. The previous equations define the discrete figure PK_n at the construction step n . They basically tell us that the global object PK_n is composed of a certain number of smaller figures that are unified by the \cup operator. As, stated in chapter 4, the construction sequence of the figure may also be represented

by the construction tree. Figure 6.2.1 shows the construction tree of a continuous curve constructed by two transformations $\{T_1, T_2\}$:

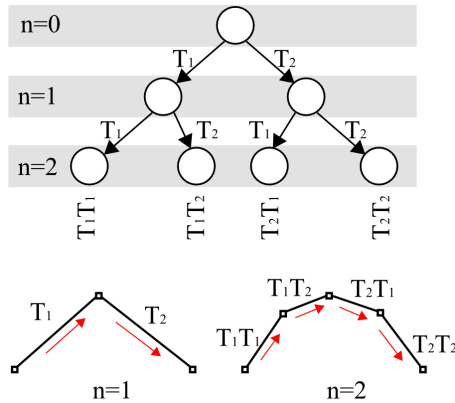


FIGURE 6.2.1. Construction tree and relating addresses

Each element of the global figure can be identified by the word composed by the transformation sequence that was employed for its construction. The addressing word is entirely composed of letters of the alphabet $\{T_1, T_2\}$, which corresponds to the list of transformations used within the IFS. In this way, we can use the word composed by the transformation sequence as a unique ID addressing one precise element. The length of the word equals the level of iteration n . In other words, the more construction steps, the longer the addressing word will be. In figure 6.2.1, at construction step $n = 2$ we have obtained 4 elements being addressed by the following four two letter words:

$$\{T_1T_1, T_1T_2, T_2T_1, T_2T_2\}$$

The number of elements N at a construction step n contained by a discrete geometric figure PK_i equals the number of transformations n_t raised to the power of the number of iterations¹ n , which means $N_n = n_t^n$. In the present example of a figure defined by the two transformations $\{T_1, T_2\}$ ($n_t = 2$), at construction step $n = 2$, we will obtain $N_2 = 2^2 = 4$ elements. If we continue the iterative construction of the figure 5 steps further, we will obtain a fairly smooth rendering of the object ending up

¹The use of the two terms “iteration” and “construction step” might be a little confusing. In fact, the term “construction step” is a more familiar expression used to describe the level of iteration in terms of IFS-modeling. Basically they mean strictly the same. However, for the annotations n will be employed, standing for number of iterations.

with $N_7 = 2^7 = 128$ elements that build up the global figure. Nota bene: The number of addresses to be generated equals N .

Although we have shown that a unique address for each element in function of the transformation sequence is provided, we need to define the neighborhood relations for further treatment. In the example, shown in figure 6.2.1, the lexicographical order of the created elements can be used. Actually, if only the indices of the transformations were considered, $\{1, 2\}$ in the present example, we would end up at $n = 2$ with the address sequence $\{11, 12, 21, 22\}$, which can be represented by a binary numeral system based on $\{1, 2\}$. In this system, each element is neighbored by two other elements; one previous and one following. This is true for all elements, except the first and the last one. Lexicographically ordered systems can basically be represented by a one dimensional ordered list L that contains n elements l_i . In this sense, the address list can be regarded as a dictionary of n entries l_i . The neighbors of an element l_i are given by l_{i+1} and l_{i-1} . Figure 6.2.2 provides a graphical representation of an ordered one dimensional list.

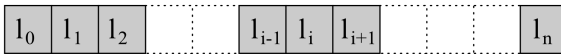


FIGURE 6.2.2. Ordered one dimensional list

Depending on the number of transformations we end up with different numerical systems. Let us consider the iterative construction of a fractal curve designed by three transformations $\{T_1, T_2, T_3\}$. Figure 6.2.3 shows us the construction tree which occurs during the construction of such a curve. The resulting addresses, at construction step $n = 2$, are again two letter words composed of elements of the alphabet $\{T_1, T_2, T_3\}$. The generated addresses present a lexicographical order, which can be compared to a ternary numerical system. Since we work on an IFS defined by three transformations, the number of elements that build up the curve at $n = 2$ is 9. Following addresses have been created at this stage:

$$\{T_1T_1, T_1T_2, T_1T_3, T_2T_1, T_2T_2, T_2T_3, T_3T_1, T_3T_2, T_3T_3\}$$

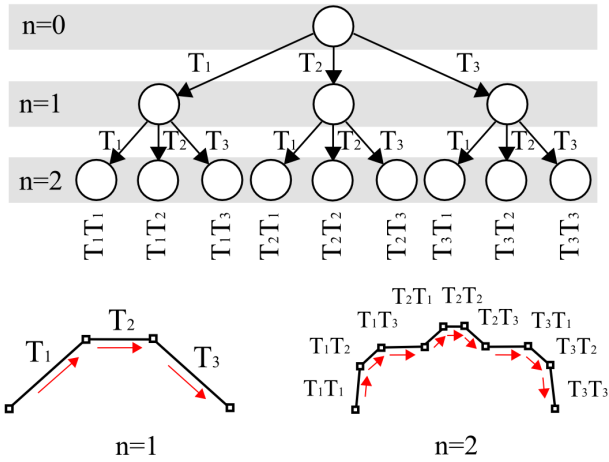


FIGURE 6.2.3. Ternary addressing system induced by the three used transformations

For the two discussed curve examples, the neighboring relations are given by the order provided in the lexicographical dictionary L .

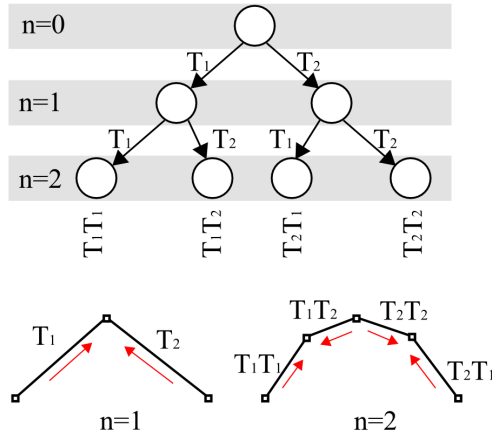


FIGURE 6.2.4. The orientation of the curve affects the addressing order.

A lexicographical order is only given if the IFS object is a regularly oriented curve. Objects of different topology can however not be represented by lexicographical systems. E.g. the two neighbors $\{T_1T_2, T_2T_1\}$ of figure 6.2.1 are not neighbors in the curve presented in figure 6.2.4.

Although the geometry of the figure remains strictly the same, the neighboring relations of the elements building up the figure has changed. As discussed in paragraph 5.3, the order and the orientation of the elements of an IFS curve may affect the order of the generated addresses. Even though the IFS curves shown in 6.2.4 and 6.2.1 lead to exactly the same Bézier curve geometrically, the generated addresses will not be ordered in the same way.

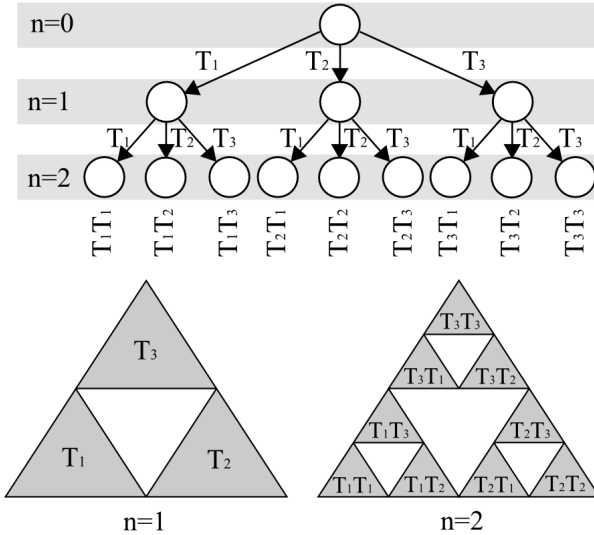


FIGURE 6.2.5. IFS-addresses of the Sierpinski triangle

After having discussed the problem of addressing the elements building up a discrete IFS-curve, we will compare the addresses of the curve constructed by three transformations (cf. figure 6.2.3) with the Sierpinski Triangle shown in figure 6.2.5, which is also an IFS-object constructed by three transformations. Although both objects are constructed by the same number of transformations, the resulting figure are totally different. While this seems to be obvious regarding the figures 6.2.3 and 6.2.5, the addressing and neighborhood relations of these two objects are totally different too. The elements building up the triangle generally have three neighbors, except the corner elements, which only have two. The elements of the curve generally have two neighbors, except the border elements, which only have one.

6.2.2. Limitations of IFS-addressing. The example of the Sierpinski triangle shows us that the addressing system depends on the topology

of the figure being designed. While linear lexicographical systems may be suitable for addressing curve objects, they are not suitable for addressing the elements of a Sierpinski triangle.

The problems relating to unsuitable order of addresses provided by the IFS-formalism are the main issues why we do not work on this IFS addressing system in a general way. Although there might be some particular cases where IFS-addressing is suitable to work with, generally, the IFS addressing will not provide the required neighborhood description.

6.2.3. Addressing faces of quadrilateral mesh figures. Within the following, the addressing problem will be examined by the case of a quad mesh figure. Two faces of a quad mesh are neighbors if and only if the two faces have one and only one edge in common. The number of neighbors in a quad mesh changes in function of the position of the face in the mesh. The number of the neighbors is also called valence of a face. Therefore, a face having four neighbors has a valence of four.

Figure 6.2.6 shows four different ways of addressing faces of a quadrilateral mesh. Row a) and b) of figure 6.2.6 shows addresses generated by the IFS-formalism. The sketch on the left column of figure 6.2.6 illustrates the subdivision logic. Per construction step, each face is subdivided into four new faces. Case a) shows a “regular” quad subdivision. The subdivision scheme of case b) has transformations that include rotation and symmetry in addition to scaling and translation. The rotation and symmetry transformations are represented by the orientations of the digits shown in the sketch.

The two bottom rows c) and d) show two ordered addressing systems which are not specifically related to IFS-addressing. Case c) shows a linearly ordered one dimensional list; d) shows an ordered bi-dimensional array. These two examples are typical addressing systems that are comfortable to work with, since they provide simple ways to access all the neighbors of every element.

The elements of the IFS figures a) and b) are not linearly ordered, due to the construction process used. Even though all the elements building up a IFS quad mesh have unique addresses, they provide not direct information about the neighborhood relations. E.g. while the figure in the first row last column shows the face element number 1 being in the bottom left corner (having two neighbors), element number 1 of the second row is actually a border element (with three neighbors). The order provided by the IFS construction algorithm has therefore to be translated into more suitable addressing order.

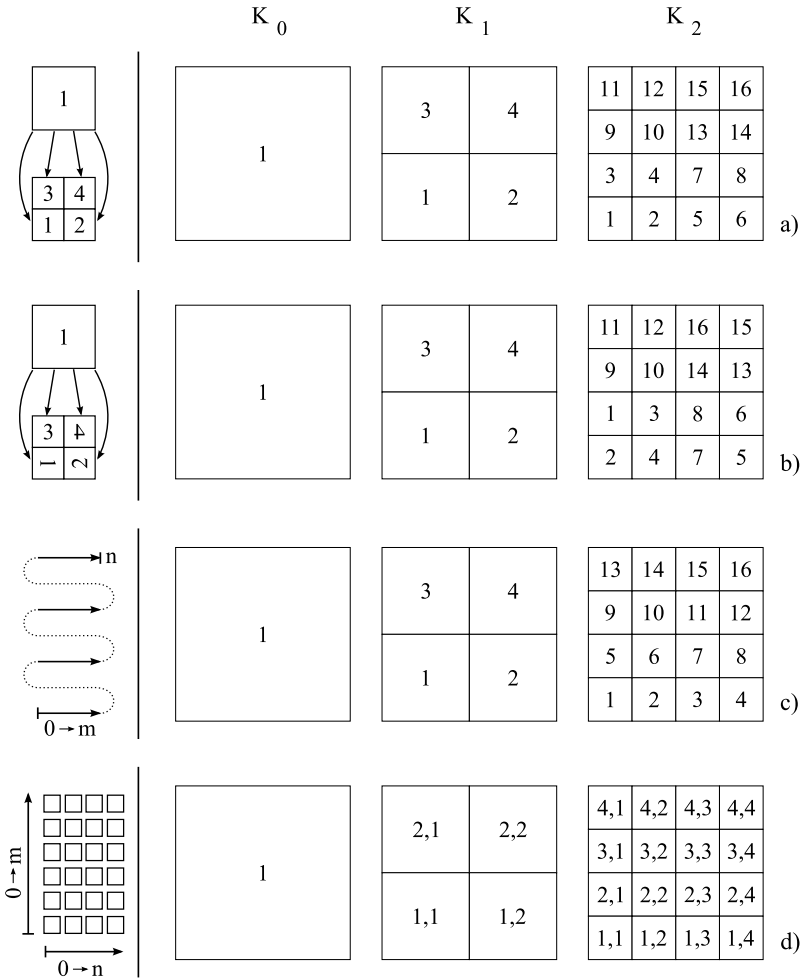


FIGURE 6.2.6. Addressing systems

Since the list of all transformations and the initial figure K is known, there theoretically exists a function which is capable of translating the order provided by the IFS into a lexicographically ordered list. This function should be able to translate the IFS-addresses of any arbitrary iterative geometric figure of arbitrary topology. The development of such a translation function is beyond the scope of the present work.

For the applications shown in 8.3 and 8.4, different methods for re-organizing the order of the quad mesh elements will be considered. The developed methods have been restricted to be used on quad mesh figures.

Before discussing the techniques allowing to re-order the elements of a IFS quad mesh, notions related to topology of quadrilateral mesh faces need to be addressed.

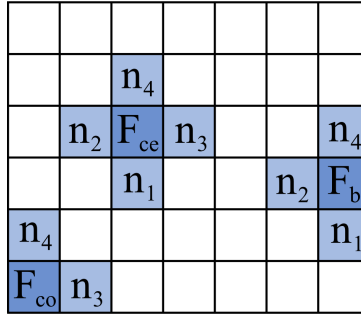


FIGURE 6.2.7. Valances of quad meshes with regards to the number of neighbors

With regards to the number of neighbors, three types of faces are defined as follows (cf. figure 6.2.7):

- Corner Faces $\{F_{co}\}$ having two neighbors.
- Border Faces $\{F_b\}$ having three neighbors.
- Center Faces $\{F_{ce}\}$ having four faces.

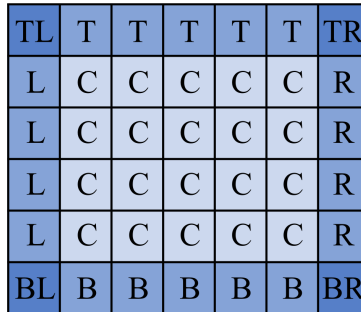


FIGURE 6.2.8. Quad mesh topology and face types (B = Bottom, L = Left, R = Right, T = Top, C = Center)

Generally, an untrimmed quad mesh has exactly four corners. Speaking of border faces (= faces with three neighbors), the following four kinds of border faces are defined: bottom, left, right and top (cf. figure 6.2.8). The reason why we have to differentiate the border faces

is that they do not have the same neighbors. E.g. a border face at the bottom has a left, a right and a top neighbor, whereas a top face has obviously no top neighbor but a bottom neighbor instead. In terms of valence, all border faces have three neighbors and therefore a valence of three. Finally, the remaining faces are center faces with the four neighbors: bottom, left, right and top. In figure 6.2.7 the different neighbors $\{Bottom, Left, Right, Top\}$ are named $\{n_1, n_2, n_3, n_4\}$.

6.2.4. Defining Neighbors Geometrically. The aim of this paragraph is to explain the geometrical techniques used to get from an unordered list of mesh faces to an ordered list, which contains information about neighbor-ship relations (as discussed in paragraph 6.2.3). Figure 6.2.10 illustrates the topic graphically, where a) is an unordered list of mesh faces, b) a one-dimensional lexicographically ordered linear list and c) a two-dimensional ordered array. Generally, we start with the case a) and will aim to get to the case b) using the geometrical methods discussed below. The passage from b) to c) is far less complicated and will not especially discussed within this section.

6	9	13	15
10	4	3	7
11	1	16	5
2	8	12	14

a)

13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4

b)

4,1	4,2	4,3	4,4
3,1	3,2	3,3	3,4
2,1	2,2	2,3	2,4
1,1	1,2	1,3	1,4

c)

FIGURE 6.2.9. Ordering mesh faces of a quad mesh figure: a) arbitrary b) linear list c) two-dimensional array

Iteratively designed figures have a faces, edge, vertex topology. Therefore an oriented edge is described by two vertices $\{v_1, v_2\}$. In the same way, a quadrilateral face is defined by four edges $\{e_1, e_2, e_3, e_4\}$. Analogous to that, a face might also be defined by four vertices $\{v_1, v_2, v_3, v_4\}$. The orientation of the edges or the order of the vertices define the orientation of the face. We will focus on meshes which have consistent mesh-vertex ordering: within a given mesh figure, all face-vertices are oriented either clockwise or counter-clockwise. In this way we assure that all faces are directly geometrically affine figures, which are oriented in the same way. Figure 6.2.10 illustrates the orientation issue of discrete polygonal figures.

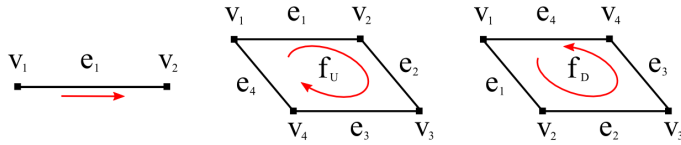


FIGURE 6.2.10. Face-Edge-Vertex order of polygonal mesh figures

With regards to quad-mesh figures, the geometry data of the four face-vertices will be compared in order to define whether two arbitrary faces of a given mesh figure are neighbors. A specific quad mesh face has up to four possible neighbors: bottom, left, right and top (cf figure 6.2.7).

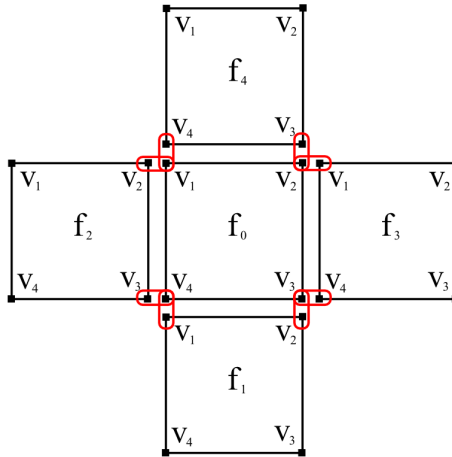


FIGURE 6.2.11. Corresponding vertices of adjacent quad mesh faces.

In order to find for example the right neighbor of the face f_0 , the geometry data of the vertices $\{v_1, v_4\}$ of the all the faces f_i will be compared successively with the data of the vertices $\{v_2, v_3\}$ of the face f_0 (cf. figure 6.2.11). If both geometrical values of the two corresponding values match, if $v_2^{f_0} = v_1^{f_i}$ and if $v_3^{f_0} = v_4^{f_i}$, the right neighbor has been found. Analogous to this, we can find the top neighbor by verifying $v_1^{f_0} = v_4^{f_i}$ and $v_2^{f_0} = v_3^{f_i}$, the left neighbor by $v_1^{f_0} = v_2^{f_i}$ and $v_4^{f_0} = v_3^{f_i}$, and the bottom neighbor by $v_4^{f_0} = v_1^{f_i}$ and $v_3^{f_0} = v_2^{f_i}$.

Generally, two conditions have to be verified for that two faces being compared are neighbors. Working with mesh faces that all have the same orientation - either clockwise or counterclockwise - makes the query for the neighbors more simple since if condition one is verified, condition two is true accordingly and vice versa, which makes the computations for the geometrical ordering much faster.

The ordering process employed to get from an arbitrary ordered mesh (cf. example *a*) of figure 6.2.9) to an ordered array (cf. example *c*) of figure 6.2.9) starts with the identification of a corner face, e.g. the bottom-left element. As discussed previously, corner faces possess exactly two neighbors, and therefore the four faces having only and exactly two neighbors will have to be found, which gives us a list of four corner faces. If the face to start ordering with is the bottom-left corner face, we would begin with the face having a top and a right neighbor. This face will become our first element of the list ; $f_{1,1}$ of example *c*) in figure 6.2.9. The second element $f_{1,2}$ of the array will be the right neighbor of $f_{1,1}$, which means $v_2^{f_{1,1}} = v_1^{f_{1,2}}$. Continuing to define successively the following right neighbors, we will end up completing the bottom row of the mesh ending at the bottom-right corner face, which does not have a right neighbor. Having completed row 1, the starting element of row 2 $\{f_{2,1}\}$ will be defined by $v_1^{f_{1,1}} = v_4^{f_{2,1}}$; being the top neighbor of the first element $\{f_{1,1}\}$. This way, the ordering process can be completed row by row. It will be complete if no top neighbor for the construction of the following row may be defined and if the number of the faces of the entry mesh equals the number of faces of resulting ordered list. The two completion conditions are true reciprocally.

We end up with a two dimensional ordered array of faces $f_{i,j}$, where $i \in \{0, \dots, m\}$ and $j \in \{0, \dots, n\}$. In the next section we will focus on how constructional elements will be derived from the geometrical elements. The resulting elements forming a piece of architecture will have to be differentiated according to their relative position in the global object. Therefore, we need to identify especially the border elements, the corner- and the center elements. With the present data structure, all faces $f_{0,j}$ are bottom border faces, $f_{m,j}$ are top border faces, $f_{i,0}$ left border faces, $f_{i,n}$ right border faces. Accordingly, the four corner faces are: $\{f_{0,0}, f_{0,n}, f_{m,0}, f_{m,n}\}$. The remaining faces $f_{i,j}$, where $i \in \{1, \dots, m-1\}$ and $j \in \{1, \dots, n-1\}$ are center faces.

The neighbors in the present data structure may be described as follows:

- Center mesh face $f_{i,j}$, where $i \in \{1, \dots, m-1\}$ and $j \in \{1, \dots, n-1\}$: $\{f_{i-1,j}, f_{i,j-1}, f_{i,j+1}, f_{i+1,j}\}$
- Bottom mesh face $f_{0,j}$, where $j \in \{1, \dots, n-1\}$: $\{f_{0,j-1}, f_{0,j+1}, f_{1,j}\}$
- Top mesh face $f_{m,j}$, where $j \in \{1, \dots, n-1\}$: $\{f_{m,j-1}, f_{m,j+1}, f_{m-1,j}\}$
- Right mesh face $f_{i,n}$, where $i \in \{1, \dots, m-1\}$: $\{f_{i-1,n}, f_{i,m-1}, f_{i+1,m}\}$
- Left mesh face $f_{i,0}$, where $i \in \{1, \dots, m-1\}$: $\{f_{i-1,0}, f_{i,1}, f_{i+1,0}\}$
- Bottom left mesh face $f_{0,0}$: $\{f_{1,0}, f_{0,1}\}$
- Bottom right mesh face $f_{0,n}$: $\{f_{0,n-1}, f_{1,n}\}$
- Top left mesh face $f_{m,0}$: $\{f_{m-1,0}, f_{m,1}\}$
- Top right mesh face $f_{m,n}$: $\{f_{m-1,n}, f_{m,n-1}\}$

6.3. Constructional Elements

6.3.1. From quad mesh faces to chamfered timber panels. Within the following, we will explain the construction of volumic elements based on planar quad faces. The main problem is that planar quad mesh faces have literally no thickness. The faces building up a mesh structure are only connected along one edge (line). Aiming to build a physical image of a free form quad mesh structure, the mesh faces will be represented by planar timber panels with chamfered edges. The geometry data describing the timber elements is represented in figure 6.3.1, where $\{v_1, v_2, v_3, v_4\} \in f$ is the initial quad face and $\{v'_1, v'_2, v'_3, v'_4\} \in f'$ is offset image of the face f ; the missing data required for the complete representation of the constructional timber element.

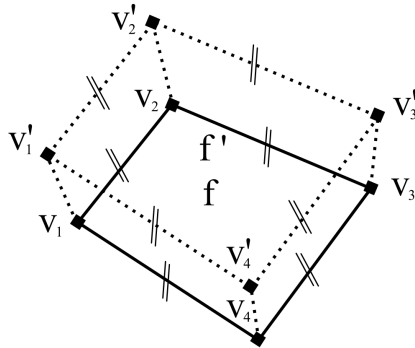


FIGURE 6.3.1. Description of a quad panel

The method to generate the data of the offset face f' may be explained on one isolated vertex, since it is the same for all the vertices $\{v'_1, v'_2, v'_3, v'_4\}$. We have chosen to work on vertex v_4 and to calculate its

image v'_4 . Figure 6.3.2 shows graphically the elements (points, planes and vectors) we work with for the calculation of v'_4 . It shows the face f_0 with its four neighbors $\{f_1, f_2, f_3, f_4\}$. A surface normal \vec{n}_0 of f_0 is defined by means of the cross product $\vec{n}_0 = \vec{v}_1\vec{v}_4 \times \vec{v}_1\vec{v}_2$. The normal vectors of the bisector planes $\{\vec{n}_{B1}, \vec{n}_{B2}\}$ can be obtained by the two adjacent planes as follows: $\vec{n}_{B1} = (\hat{n}_1 + \hat{n}_0) \times \vec{v}_1\vec{v}_4$, $\vec{n}_{B2} = (\hat{n}_3 + \hat{n}_0) \times \vec{v}_4\vec{v}_3$. Where, the first vector used for this cross product is the sum of the two unitized normal vectors $\{\vec{n}_0, \vec{n}_1\}$ being collinear to the bisector plane B_1 . The second vector is defined by the common edge $\{v_1, v_2\}$, that is also collinear to B_1 .

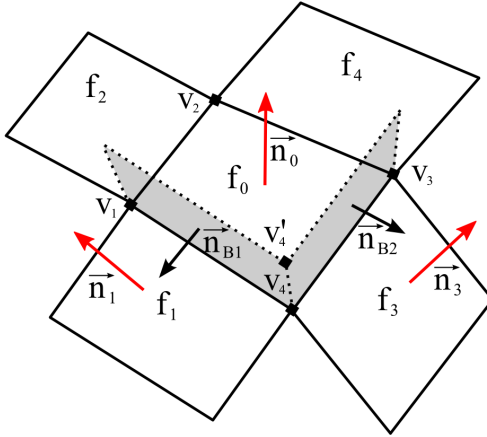


FIGURE 6.3.2. Face offset: generation of the vertex v'_4

Since we know that v_4 is element of both bisector planes, and since we have obtained their normal vectors, the two bisector planes $\{B_1, B_2\}$ are fully defined. Point v'_4 is part of the intersection line V_4 of the two bisector planes. The parametric representation of V_4 is:

$$V_4 = v_4 + t(\vec{v}_4) \text{ where the unit vector } \vec{v}_4 \text{ may be defined by } \vec{v}_4 = \frac{\vec{v}_4\vec{v}'_4}{\|\vec{v}_4\vec{v}'_4\|}.$$

Since V_i is the intersection of the two bisector planes, the direction of the vector $\vec{v}_4\vec{v}'_4$ is perpendicular to both normal vectors $\{\vec{n}_{B1}, \vec{n}_{B2}\}$ and corresponds therefore to the direction of their cross product:

$$\vec{v}_4 \equiv \frac{\vec{v}_4\vec{v}'_4}{\|\vec{v}_4\vec{v}'_4\|} \equiv \frac{\vec{n}_{B1} \times \vec{n}_{B2}}{\|\vec{n}_{B1} \times \vec{n}_{B2}\|}$$

The missing value of the parameter t_4 in $v'_4 = v_4 + t_4 \vec{v}_4$ might be obtained by following expression $t_4 = \frac{d}{\vec{n}_0 \bullet \vec{v}_4}$. Where d stands for the offset distance, which is in function of the thickness of the timber panel used. Hence, we have completed the demonstration of how to calculate the vertex v'_4 using the quad face f_0 and its two neighbors f_1 and f_3 . The vertices $\{v'_1, v'_2, v'_3, v'_4\}$ describing the offset face f' of any thickness d can be generally described by:

$$v'_i = v_i + t_i \vec{v}_i$$

6.3.2. Border conditions for Border elements. The explanations given above describe the generation of constructional volume elements from planar quad mesh faces. For the computation of the set of vertices $\{v_1, v_2, v_3, v_4, v'_1, v'_2, v'_3, v'_4\}$ describing the geometry of the timber panels, the face being thickened f_0 require all four neighbors $\{f_1, f_2, f_3, f_4\}$. In a quad mesh figure, the required neighbors only exist for center mesh faces (cf. figures 6.2.7 and 6.2.8).

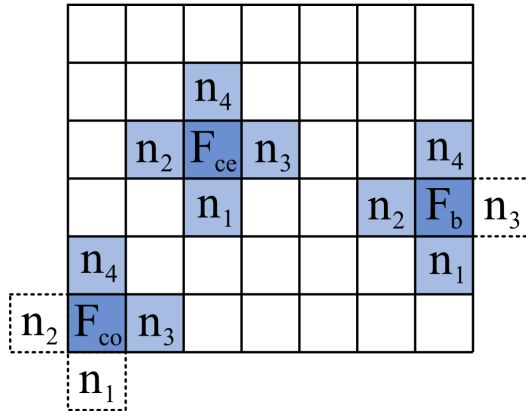


FIGURE 6.3.3. Imaginary neighbors of border and corners mesh faces

Contrary to center faces, border and corner mesh faces present only respectively three and two neighbors. Missing “imaginary” neighbors need to be defined (cf. figure 6.3.3). Therefore, additional conditions need to be formulated for the design of the border elements, that will be depending on conditions given by the architectural context.

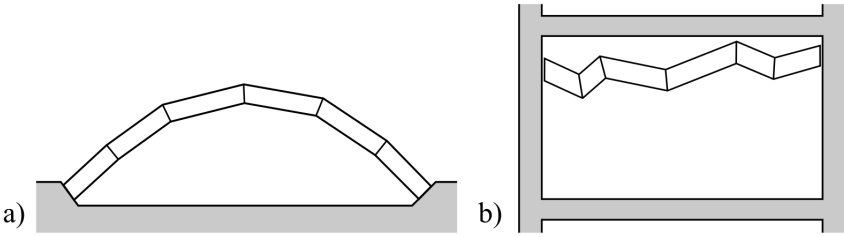


FIGURE 6.3.4. Applications of free form quad meshes:
 a) bearing structure b) suspended ceiling

In the case of free form quad mesh surfaces that are applied for architectural use, the border conditions are generally dependent on the context. Figure 6.3.4 shows sketches of two example applications, where case a) represents a load bearing shell structure and b) a suspended free form ceiling. In case a), the border faces must be designed such that the load is correctly transmitted from the shell structure to the supports. In such a case, the chamfer cut of the border panel facing the support needs to be right-angled to the support surface. In case b), we have a mesh geometry being used to describe a ceiling that goes from one side of the room to the other. In this case, the border elements' cut off needs to be parallel to the delimiting wall surface.

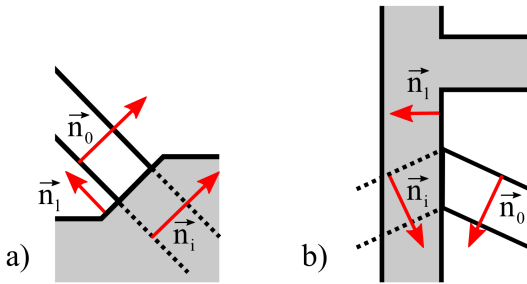


FIGURE 6.3.5. Normal vectors of the border mesh face, the limit surface and the imaginary neighbor

Based on these two examples, the definition of the missing “imaginary” neighbor faces will be explained. Actually, the imaginary neighbors do not need to be expressed in form of a quad face defined by four vertices. The actual data we need of these imaginary neighbors are their normal vectors allowing us to complete the computation of the timber

panel geometry data. Figure 6.3.5 shows both cases more in detail; focusing on the border element. The border panel is illustrated by means of the its normal \vec{n}_0 . The imaginary neighbor is also indicated by its normal vector \vec{n}_i .

For both examples a) and b), the same calculation method of \vec{n}_i is used. \vec{n}_i is the image of \vec{n}_0 mirrored by the limit plane. In the case b) the wall will be considered as the limit plane. In case a), the support surface is considered as the limit plane. The limit surface is represented by its normal vector \vec{n}_l . The normal vector of the imaginary neighbor \vec{n}_i may be defined by:

$$\vec{n}_i = \vec{n}_0 - 2\vec{n}_l (\vec{n}_l \bullet \vec{n}_0)$$

Nota bene: If, and only if, the limit surface is perpendicular to the border face, the imaginary neighbor's normal vector equals the border face's normal vector: $\vec{n}_i = \vec{n}_0$.

6.3.3. Examples of thickened planar quad meshes. This paragraph briefly illustrates the topic of thickening regular quad mesh surfaces by a series of different designs. As shown in figure 6.3.6, the entry geometry - presented on the left side of the figure - has no thickness. On the right side of the same figure, we have the resulting geometry where each element represents the geometry data of a chamfered timber panel. The top-border panels have been cut off exactly horizontally. With regards to the left- and the right-border elements, the cut off of the elements is right angled.

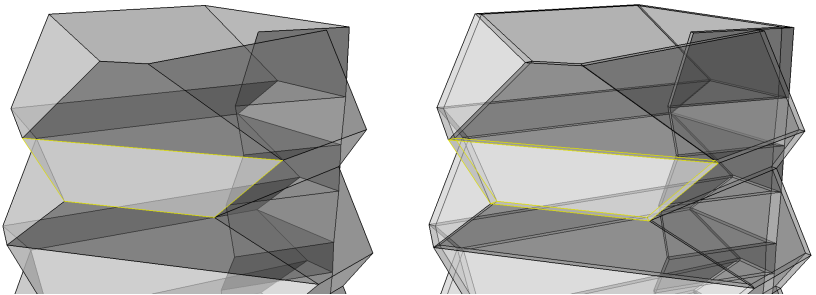


FIGURE 6.3.6. Thickening of a folded plate structure

The second example, cf. figure 6.3.7, shows the same thickening method applied to a different quad mesh entry surface. This second example illustrates that the method is independent on the entry mesh geometry and the amount of thickening d .

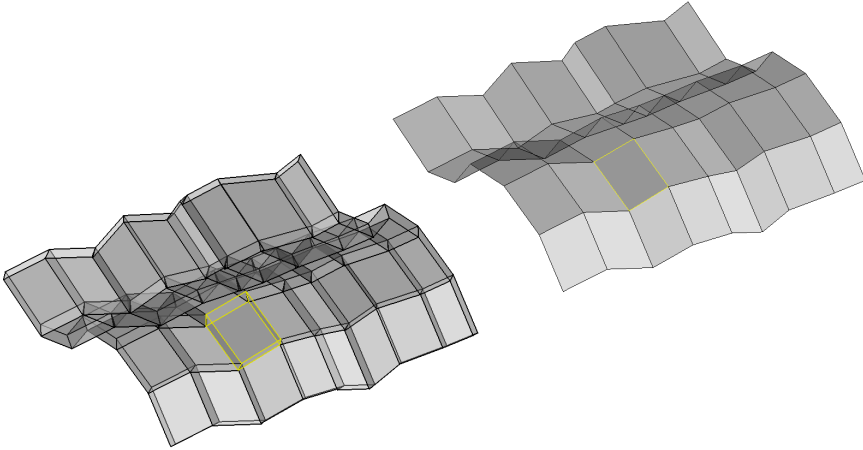


FIGURE 6.3.7. Thickening of a planar quad mesh surface

6.3.4. Conical Meshes. Above, we have established a deterministic method for thickening planar quad mesh faces, which allows differentiating the center faces from the border faces. Unfortunately, the method studied works only on specific meshes verifying certain geometrical conditions. As shown by Pottmann and Wallner [PW08], the condition for the generation of consistent offset surfaces is restricted to so called conical quad meshes. A conical mesh can be described by the geometry of a vertex with its four coinciding edges. The geometrical constraint required for a vertex to be conical may be expressed by following angle criterion: $\alpha + \gamma \equiv \beta + \delta$ (cf. figure 6.3.8). Conical vertices are specific to quad meshes, which have vertices of valance four. Conical vertex offsets only exist on inner vertices of a quad mesh figure, as the border vertices have a valance of three.

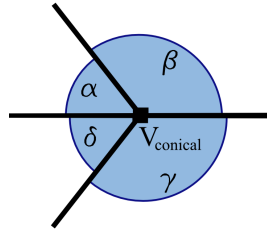


FIGURE 6.3.8. Angle criterion according to [PW08]

Generally, the surface method studied in 4.3 does not provide conical meshes. However, the generation of conical meshes using the proposed mesh design method is possible, as shown by the examples presented in figure 6.3.6 and figure 6.3.7. If applied to non-conical quad meshes, the thickening method discussed above provides either overlapping constructional parts or gaps in the nodes.

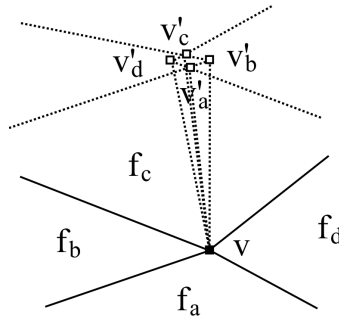


FIGURE 6.3.9. Offset problem on non-conical vertices

Figure 6.3.9 illustrates the offset issue of non-conical vertices. It represents a non-conical vertex v with its four adjacent faces $\{f_a, f_b, f_c, f_d\}$. If each of the faces is thickened using the method described previously, the resulting vertices v'_i would generally not lie in the same point. Each of the thickened corner vertices $\{v'_a, v'_b, v'_c, v'_d\}$ has a different position in space, except if, and only if, the vertex v is conical.

6.3.5. Offset of non-conical meshes. The developed surface method (cf. section 4.3) is not constrained to produce conical meshes only. Therefore the offset method has to provide solutions for the generation of constructional elements of non-conical quad mesh figures.

Since the applications are built out of timber panels, which are connected along their edges, the actual node is not so important compared to wire-frame or reticulated structures. The solution shown below (cf. figure 6.3.10) proposes to void the vertices such that the newly generated offset face f' is no longer a quadrilateral but an irregular octagon.

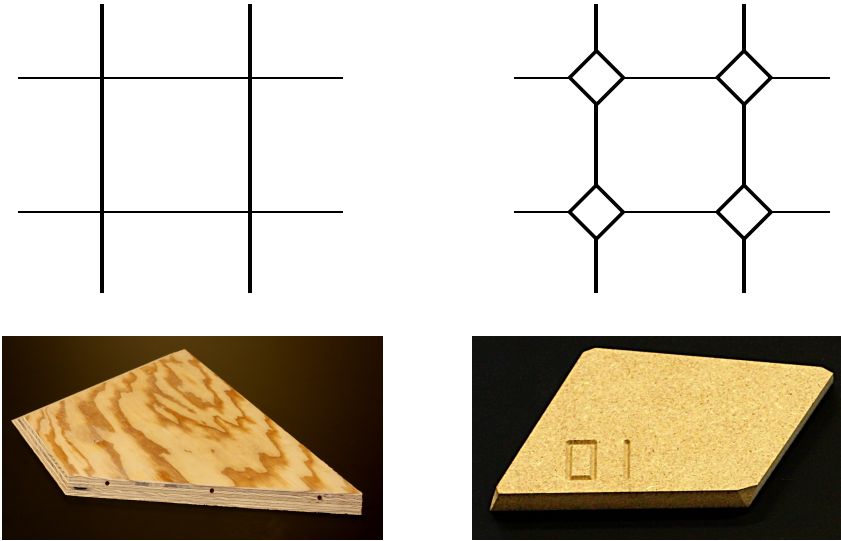


FIGURE 6.3.10. Resulting constructional elements of conical and non-conical meshes respectively

This solution has been applied on the timber plate structures presented later in section 8.4. It implies that one side of the resulting physical object corresponds exactly to the geometric figure while the other side presents “open” nodes, conducting to free-form surfaces with different front and back sides. The decision of voiding the nodes has several consequences on the designed free-form object.

Within the scope of our work, if applied on a large scale building as proposed in case a) of figure 6.3.4, where the surface describes a bearing dome structure, the front side may represent the inner side, being structure and finishing at the same time, while the back side would typically be covered by supplementary layers such as thermal insulation, coatings, weather sealing, cladding, etc. In case b) of figure 6.3.4 the proposed application is a suspended ceiling, where the back side is hidden.

While there exist a certain number of applications where two different sides may be accepted, the design of structures of which both sides are

directly visible - as for example the origami sculpture shown in 11.3.2 - require the quad mesh to be conical.

6.3.6. Notes on offset meshes. Actual CAD-software provide different mesh offset methods. We have found that existing offset methods do not integrate all the requirements in order to be suitable for the presented application, which require:

- Two corresponding mesh faces $\{f, f'\}$ must be exactly parallel
- The offset distance d between all mesh face pairs $\{f, f'\}$ must be constant
- Two correspondent edges must be collinear
- Different offset conditions have to be applied to border vs. center mesh elements

In order to meet these constraints, the offset method presented above has been developed.

6.4. Generic Detailing

Once the volume describing the base geometry of the constructional timber element has been defined, the joint details have to be designed in function of the used assembly fasteners. This task can later be automated, which will briefly be addressed in section 6.5. The issue of detailing will be discussed by the example of planar quad mesh surfaces. The joining of planar quadrilateral timber panels will be realized along the edges by a series of linearly distributed fasteners. The exact form of the joints as well as the type of fasteners used is completely dependent on the conditions to which the free-form surface will be exposed. On the one hand, the joint might have to transmit loads from one element to the other, on the other hand, the joint may also have to comply with weather sealing constraints. However, these two scenarios show that, depending on the type of application, the actual detail of the joint might be completely different.

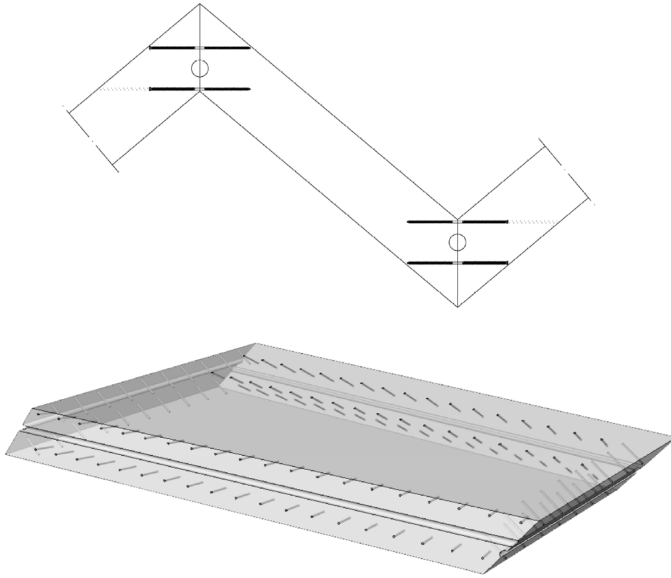


FIGURE 6.4.1. Example of a set of linearly arrayed fasteners along the edges of a quadrilateral timber panels

6.4.1. Linear fastener distribution. Within this section, we will try to focus on a more general notion of joining timber panels along their edges. The examples serving to illustrate the detailing process are always subject to a specific context, which will not be particularly questioned within the scope of this section.

Before explaining more in detail the technical aspects, we propose to illustrate the topic by an example. Figure 6.4.1 shows on top a section of a connection detail proposed by the civil engineer Marcel Haasis. As stated above, we will not explain exhaustively the actual development of the shown detail.

The present detail was designed to transmit loads and forces from one panel to the next, while providing a certain bending rigidity and avoiding relative displacement between two panels. The design of the joint detail will be taken as a given condition. For interested readers, detailed information about the employed detail can be found in [HW08b].

The actual goal consists in describing a given detail by a set of parameters, which then could be applied on all four sides of every constructional element. The chosen detail necessitates the manufacturing of

the constructional timber elements' edges. The generic detail could be described as follows:

First a slot (to receive the key) has to be grooved along each edge of the panel. Second, a series of holes will have to be drilled, that will receive the screws fastening two panels together. The screws are positioned within a certain interval that is depending on the type of screw and the wood species used. Typically, the interval is calculated by the civil engineer and assumed being constant. Further, the screws have to maintain a minimal distance to the edge. Finally, the direction of the screws has been defined to be perpendicular to the chamfer surface.

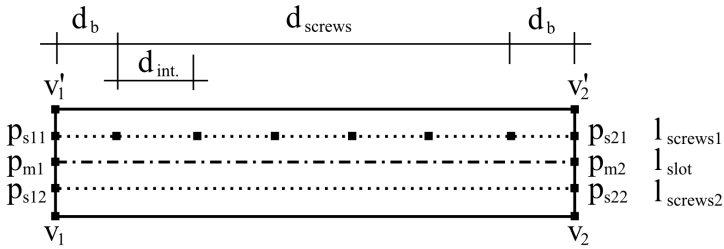


FIGURE 6.4.2. Parametrical representation of the detail of one side of a quad panel according to figure 6.4.1

Figure 6.4.2 illustrates the used parameters, that are explained as follows:

- The chamfer surface is described by the vertices $\{v_1, v_1', v_2, v_2'\}$ obtained by the thickening process.
- l_{slot} defines the line of the slot (defined by $\{p_{m1}, p_{m2}\}$); computed by $av_1 + bv_1'$ and $av_2 + bv_2'$ respectively, where the coefficients $\{a, b\}$ define the ratio of the height on which lies the slot line
- $\{l_{screws1}, l_{screws2}\}$ are the lines on which the screws will be placed. They can be computed similarly to the way in which the slot line l_{slot} has been calculated.
- Distance to the border (for the screws) d_b
- d_{screws} available distance to distribute the screws:

$$d_{screws1} = \|\vec{p_{s11}p_{s21}}\| - 2d_b, d_{screws2} = \|\vec{p_{s12}p_{s22}}\| - 2d_b$$
- $d_{int.}$ interval distance of the screws

The interval of the screws $d_{int.}$ is related to the length of the screw line l_{screws} . To determine $d_{int.}$ we need to compute the number of screws being placed on l_{screws} . Therefore, the available distance to distribute the screws d_{screws} will be divided by the nominal distance between the screws, required by the civil engineer; e.g. every 15 centimeters. The result is then rounded to the next integer, which gives us the number of screws N_{screws} . Finally, $d_{int.} = \frac{d_{screws}}{N_{screws}}$ defines the interval of the screws being placed.

In the present example, shown in figure 6.4.2, the number of fasteners vary in function of the length of the edge. However, the parametric description allows adapting the number of fasteners in function of the chamfer surface's geometry. Since the chamber surface lies in the bisector plane between two elements, the chamfer surfaces of both elements are geometrically identical. Therefore, the detailing for the fasteners connecting the two elements may be done on each element individually.

The reason why pre-drilled holes have shown their advantage is that the montage can be undertaken faster and with a greater precision, which is of high importance for the accuracy of the perfectly fitting parts.

6.4.2. Fasteners for joining timber panels. The type of fasteners used depends on the type of application. The joint detailing of the example discussed previously had to comply with certain load bearing conditions. Joint detailing for interior design may use other types of fasteners. Other joint details may have to deal with regards to weather sealing. Within this paragraph, a limited series of fasteners for joining timber panels will be discussed.



FIGURE 6.4.3. Mechanical fastener for joining panel at various angles

A series of such fasteners is shown in figure 6.4.4. The dashed lines indicate the position of the tool axis used for the machining of the detail. In this series of drawings, the fasteners have been drawn in situations of angled panels; negatively and positively angled panels. All the fasteners shown in figure 6.4.4 are depending on the geometry of the panels. For the examples {a),d),e),f)}, the panel's cut off lies in the bisector plane. In such cases, the thickening method presented in 6.3 can be employed for the computation of the components geometry. For the examples {b),c)}, a different cut off is used depending on the fastener type and the angle between two neighboring elements.

The assembly of two panels using hard wood dowels situated in the bisector plane can be realized only up to a certain angle, as shown in case e). If the two chamfered panels were right-angled, a dowel joint becomes rather difficult to realize, since the remaining material holding the dowel in place gets very small. One possible solution to this problem might be to chamfer the panels not at their bisector plane but rather to cut them as applied in case b) of figure 6.4.4.

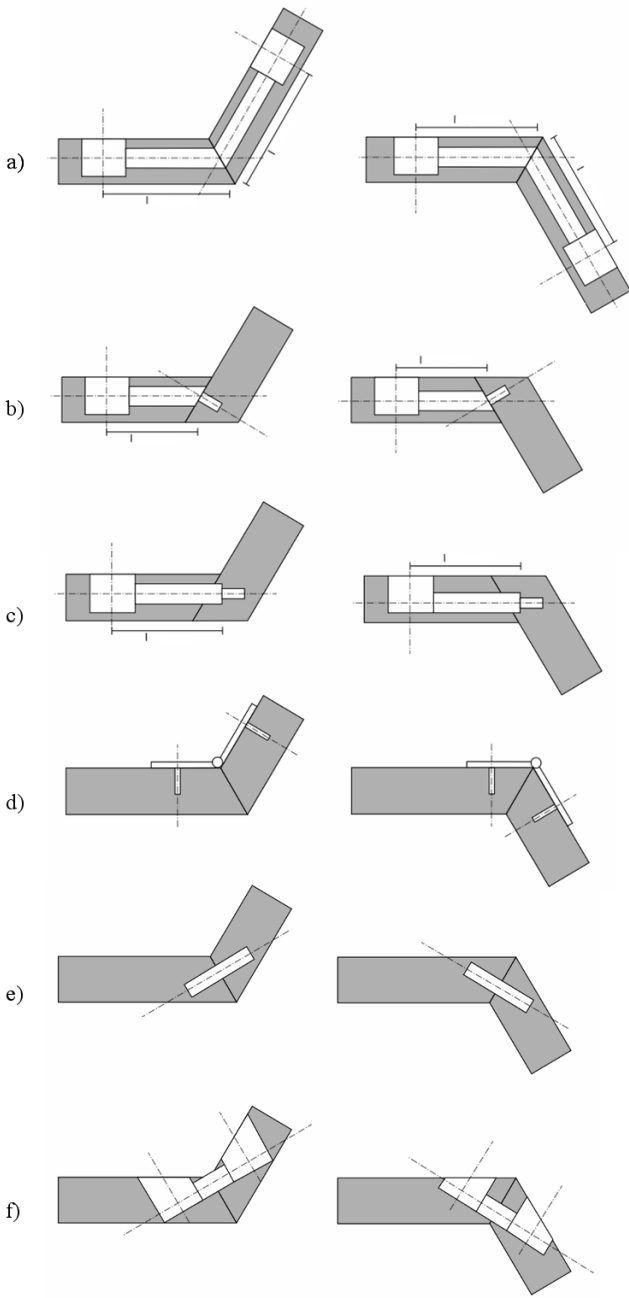


FIGURE 6.4.4. Different fasteners for the assembly of timber panels: Hinge joints {d}, hard wood dowels {e}, mechanical fasteners {a,b,c,f}

With regards to assembly, two types of fasteners will have to be distinguished:

- Exterior fasteners: They are installed after the panels being joint have been brought into position. Exterior fasteners are e.g. hinge joints like shown in case d) in figure 6.4.4 or screws type fasteners used for the detail shown in figure 6.4.1. Hinge joints (cf. figure 6.4.4 d) have been employed for the construction of a temporary chapel building made of massive timber panels [BW08a, BW06a, BW06b].
- Interior fasteners: They are installed in one or both pre manufactured sides of the panels being joined before the elements are brought into position. Image 6.4.3 shows an example of a mechanical dowel type fasteners. Slot/key connections and hard wood dowels also belong to this category of fasteners.

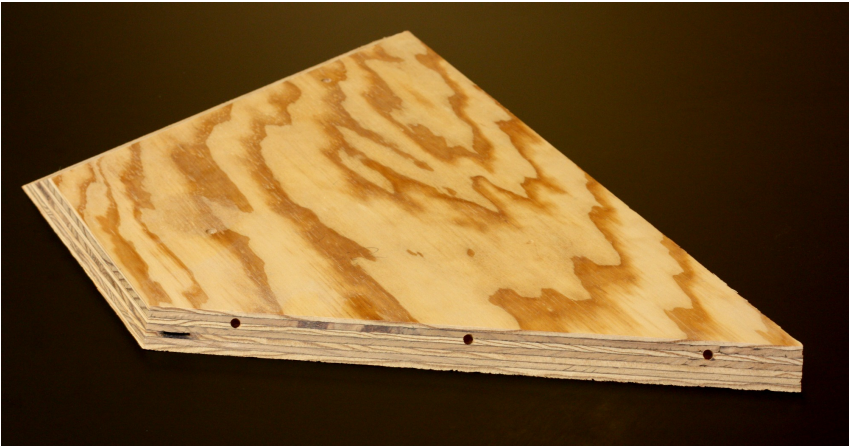


FIGURE 6.4.5. Panel with pre-drilled holes for hard wood dowels

Figure 6.4.5 shows a machined panel with pre-drilled holes for hard wood dowels. Note that the holes have intentionally be placed above the middle of the chamfer surface. A representation of the section through the dowel can be seen in case e) of figure 6.4.4.

6.4.3. Montage of quad panel structures. The type of fasteners used conditions the assembly process of the structure being built. The way in which the individual constructional elements are brought into position before being fastened depends on the fastener type. E.g. dowel connections require the pieces to be slid along the axis of the dowel. The implied handling demands a minimum of free space around the panel in order to permit the sliding manoeuvre. Generally, assembling two panels by dowels does not present any particular problems; one shifting direction being defined. Contrary to this, the joining of a panel to two neighboring elements simultaneously is impossible, since two distinct shifting directions provide a overdetermined system.

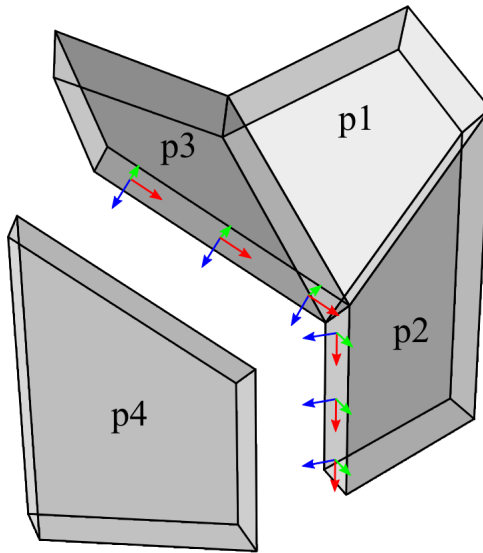


FIGURE 6.4.6. Montage problem with dowel type fasteners

Figure 6.4.6 shows the montage problem of dowel type connections. After the assembly of the panels $\{p1, p2, p3\}$, it is impossible to insert panel $\{p4\}$ since it would necessitate to be slid along two different directions simultaneously. One direction is given by the blue axis of $\{p2\}$ while a second direction is defined by the blue axis given by the fasteners towards panel $\{p3\}$.

In the particular case shown in figure 6.4.6, a solution exists nonetheless: The structure may first be assembled in rows $\{p1, p2\}, \{p3, p4\}$, which later will be assembled together. This kind of pre-montage works

Therefore, the robustness for the montage handling may become the determining criteria for the design of the type A joint.

The example of the row montage was addressed within this section with regard to detailing. Remarkably, the assembly logic influences the connection details. The established data structure allows the definition of different details towards the top-, bottom-, right- and left neighbors. Moreover, details for inner connections (joining two center panels) may be considered separately from the border connections joining the free-form structure to its external context (support joints, wall joints, etc.).

6.5. Discussion

The topics addressed within this chapter allow us to structure the geometry data provided by the geometric design method (cf. chapter 4) and to apply further post-treatment in order to obtain a description of a coherent set of constructional parts. The representation of the geometric elements has been reduced to a set of parameters and variables allowing to automate the generation of the manufacturing data of every single constructional element individually. Although all the elements building up the iteratively designed figures are topologically of the same type, the border conditions provided by the context are demanding different treatments of the details. Within this chapter, several examples complying with these constraints have been presented.

Being conscious that the topic was mainly explained by the example of planar quad mesh structures, the presented methods are generally applicable to other applications as for example:

- The generation of the workshop plans serving for the construction of a Bézier vault structure built out of timber planks is for example described in section 8.2.
- Addressing and generation of workshop plans for the 256 elements composing a smooth B-spline shell, which is addressed in 8.3.
- Automated generation of workshop plans for the 10'000 elements (including addressing, generation of the component's geometry and detailing) of a large scale grid shell is mentioned in the appendix.

The aforementioned examples illustrate the potential of different aspects related to the topic of automated generation of workshop plans.

CHAPTER 7

Integrated Manufacturing

7.0.1. Scope of this chapter. This chapter deals with questions related to the production of constructional elements building up iteratively designed figures. The conditions given by the manufacturing process constrain the geometric model. Some of the manufacturing conditions are related to material properties while others are implied by the machining process employed.

This chapter will discuss a series of examples showing how the data describing the constructional elements (obtained by means discussed in chapter 6) will be translated into a set of machine instructions with regards to integrated manufacturing. Finally, the goal of the proposed methods is to obtain a set of constructional elements ready for assembling.

7.0.2. Integrated manufacturing techniques. In the field of timber construction, multipurpose 5-axis CNC-machines are nowadays widely employed for the production of constructional timber elements. These machines are relatively large scale routers which can operate with more or less specific wood working tools. The dimensions of these CNC-machines range from 2 by 3 meters up to 30 by 5 meters, allowing to produce important carpentry work pieces. Generally, machines of this type provide automatic tool change permitting the execution of saw cuts, tapping operations or routing machining on the same machine, to name just a few.

7.1. Mapping Problem

The elements constituting iteratively designed figures are situated in the model space, which is generally \mathbb{R}^3 . In order to be machined, the elements need to be re-mapped with regards to the working coordinate system of the CNC-machine and with regards to the dimensions and the orientation of the raw material being machined.

Usually, CNC-machines possess several coordinate systems that are either defined globally at the machine's axis extends, relatively to the

tool tips position, or relatively to an temporary defined coordinate system. Further, different origins can be defined: absolute machine zero, workpiece zero, positioned or “temporary” zero point. Without going further into the various combinations of origins and coordinate systems of the machine, we will briefly address the method used for mapping one part of the iteratively designed geometry, assumed to have an arbitrary position and orientation in the model space \mathbb{R}^3 , to the origin of the model space. Later, the model space coordinate system will be mapped to the CNC-machine’s working coordinate system.

Therefore, an object coordinate system will have to be designed for each element of the iteratively constructed figure. Figure 7.1.1 shows an axonometric view of a quad mesh situated in the design space \mathbb{R}^3 . The model space is represented by the world coordinate system $\{O_w, X_w, Y_w, Z_w\}$. Face f has the object coordinate system $\{O_o, X_o, Y_o, Z_o\}$, where $\{v_1, v_2, v_3, v_4\}$ lie in the plane $X_o Y_o$ and Z_o is collinear to the normal vector of f . The origin of the object coordinate system is v_1 , the orientation of the plane $X_o Y_o$ is defined by $X_o = \frac{v_1 v_2}{\|v_1 v_2\|}$.

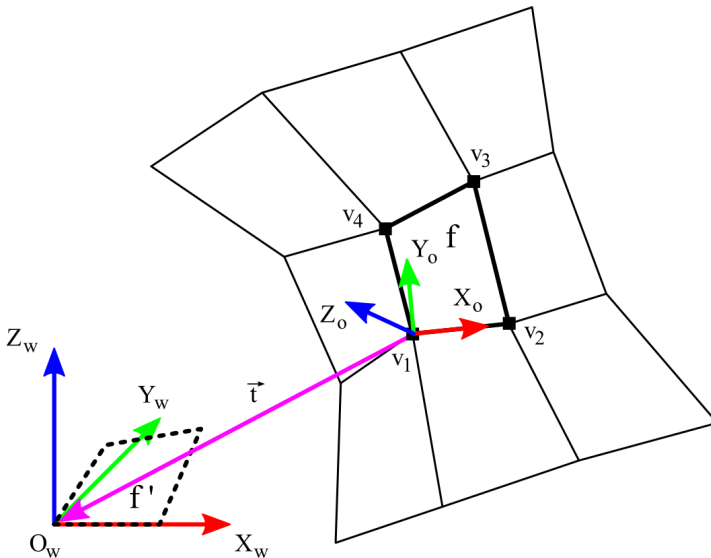


FIGURE 7.1.1. World and object coordinate system

Mapping face f to its image f' can be described by a set of three affine geometric transformations. First, the translation $\vec{t} = \overline{v_1 O_w}$ brings f to the origin of the design space (cf. figure 7.1.1) providing the image

$f'_{\vec{t}}$. Two rotations $\{A_\alpha\alpha, A_\beta\beta\}$ allow to map the object coordinate system collinearly to the world coordinate system (cf. figure 7.1.2). The first rotation α has A_α as rotation axis, while the second rotation β has A_β as rotation axis. The rotation axis A_α is defined by the cross product of the two Z-axis: $A_\alpha = O_o + t(Z_w \times Z_o)$. The rotation axis A_β is defined by world Z-axis: $A_\beta = O_o + t(Z_w)$. The first rotation brings $f'_{\vec{t}}$ into the $X_w Y_w$ -plane providing $f'_{\vec{t}\alpha}$, while the second rotation orients $\{v_1, v_2\} \in f'_{\vec{t}\alpha}$ parallelly to the world X-axis X_w providing the completely mapped face $f'_{\vec{t}\alpha\beta}$ (cf. figure 7.1.2).

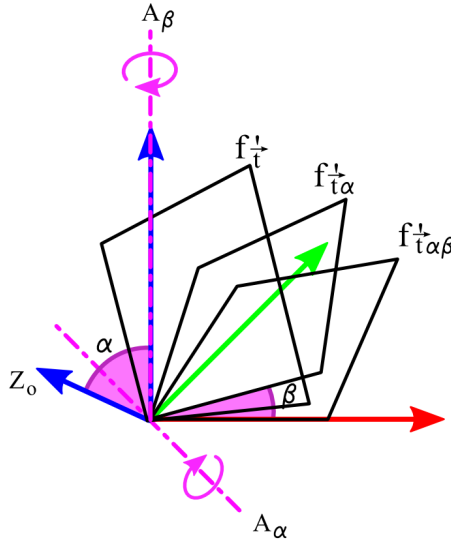


FIGURE 7.1.2. Rotations mapping the object coordinate system collinearly to the world coordinate system

The mapping method, illustrated by the example of a quad face, can be applied on any element of a given IFS-figure, if and only if, an object coordinate system can be defined for the element being mapped. The parameters defining the object coordinate system influence the position/orientation of the resulting mapped element. Actually, the object coordinate system used in the previous example is aligned along the edge $\{v_1, v_2\}$. However, it would have been possible to orient the object coordinate system e.g. aligned along the edge $\{v_1, v_4\}$.

Once the parameters of the object coordinate system have been specified generically, the generation of individual object coordinate systems

(and hereby the mapping) can be computed individually for each and every single element of a given figure.

The parameters defining the final orientation of the constructional element are often determined by external conditions. E.g. the proportions and dimensions of the manufacturing area may condition the orientation of an element being machined. Another factor influencing the orientation of the constructional elements is for instance wood grain direction. For structural applications, a load bearing element needs to be oriented precisely with regard to the grain.

7.2. Packing Problem

The packing problem is extending the mapping problem, which brings each constructional element to the origin of the design space. Packing addresses the problem of how the pieces are placed within a block of raw material. Within the following, the packaging problem is strictly related to multi-part manufacturing. In the case of timber plate constructions, the raw material may be a massive timber block panel out of which several constructional elements will be machined.

The dimensions of the raw material will be describing a container, while the set of constructional elements will be considered as the goods being packed into the containers. The goods have to be packed such that they geometrically do not overlap. Depending on the size, shape, and number of goods, the packing has to be optimized, such that the quantity of containers is minimized while minimizing the empty space left in the containers.

Today, there exist no optimal and deterministic solution to this problem. We will not provide the optimal solution either. However, depending on the way in which the packing is done, the number of containers, and therefore the quantity of raw material used may vary from the simple to the double.

Several methods have been studied for more regular conditions, like circle packing, or square packing in a 2D-container (cf. [CS93]). In our case, the goods are far less regular objects representing three dimensional polyhedral volumes, which is not advantageous for finding a solution to the packing problem. However, the fact that the constructional elements are all of the same thickness, allows us to reduce the problem to a pseudo 2D packing situation.

7.2.1. Description of the goods. In order to describe the 3D-elements, which are going to be packed in a 2D-plane, we need a pseudo 2D-description for every element. This 2D-description will be expressed by a polygon circumscribing the geometry of the constructional element and the demanded space for the tool to machine the element (cf. figure 7.2.1).

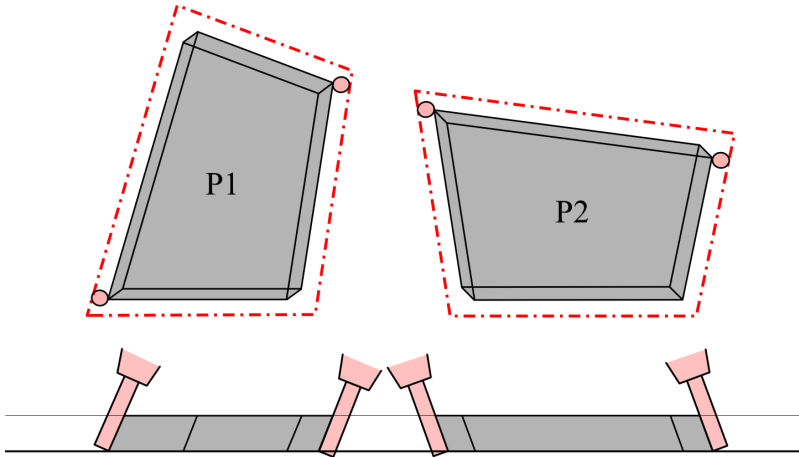


FIGURE 7.2.1. Pseudo 2D-description of two constructional elements.

Figure 7.2.1 illustrates two work pieces with their corresponding 2D-polygons; plan and front view. Depending on the machining strategy and depending on the tool used, the demanded free space, which is necessary to manufacture the element, is going to be greater or smaller. E.g. machining the pieces with a circular saw blade (cf. figure 7.3.3) will necessitate more space for tool manoeuvres than machining the same piece with a flat tipped finger tool (as shown in figure 7.2.1).

The integration of CAM parameters - such as machining strategy and tool geometry - allows to create a description of the goods. Hence, the proposed packing solution is able to take into account parameters from the level of machining. Problems of packing and machining are generally treated separately, which is a major reason why no general packing solutions exist for 5-axis multi-part machining at present.

7.2.2. Packing process. Once the circumscribing polygon (red dash-dot line cf. figure 7.2.2) has been defined, the actual packing process may be undertaken. Note, for orientation reasons with regards to wood grain

direction discussed in 7.1, the goods are not allowed to be turned or re-oriented in any sens, which is constraining the packing possibilities to operate on translations only. While, this constraint may help to define the packing process, it actually increases the number of containers needed to hold all of the elements.

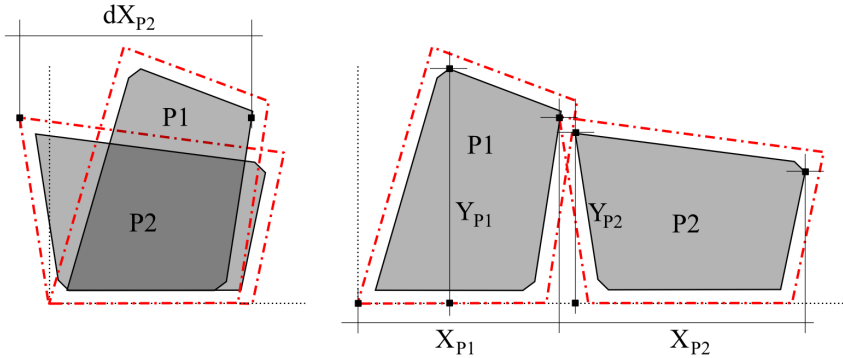


FIGURE 7.2.2. Packing P2 with regards to P1

In addition to the circumscribing polygon, a workpiece describing polygon is going to be defined (gray area on figure 7.2.2). The packing is done by rows. Each constructional element is translated along the X-axis. The translation is defined by dX_{P2} . This way, the packing allows overlaps of circumscribing polygons, since they contain only waste material. Contrary this, the workpiece describing polygons must not overlap with any other parts.

Once the element being packed has been positioned in the container, its needed space in X- and Y-direction is computed respectively and defined as $\{X_{P_i}, Y_{P_i}\}$. This data is then used for the construction of successive packing rows. One row may contain as many elements as $\sum X_{P_i} \leq X_{raw}$ is verified, where X_{raw} is the width of the raw material and $\sum X_{P_i}$ the width of all the elements already placed in the actual row. Note, $\sum X_{P_i}$ is always computed within the scope of one single row.

If $\sum X_{P_i} > X_{raw}$, a new packing row will be started. The following element will be packed at $\{X = 0, \sum Y_{row_i}\}$. Y_{row_i} is the biggest Y_{P_i} value of a row. If $\sum Y_{row_i} > Y_{raw}$, the packing of a container - one raw material panel - has been completed. Remaining elements, which haven't been packed yet, will be packed in a following container.

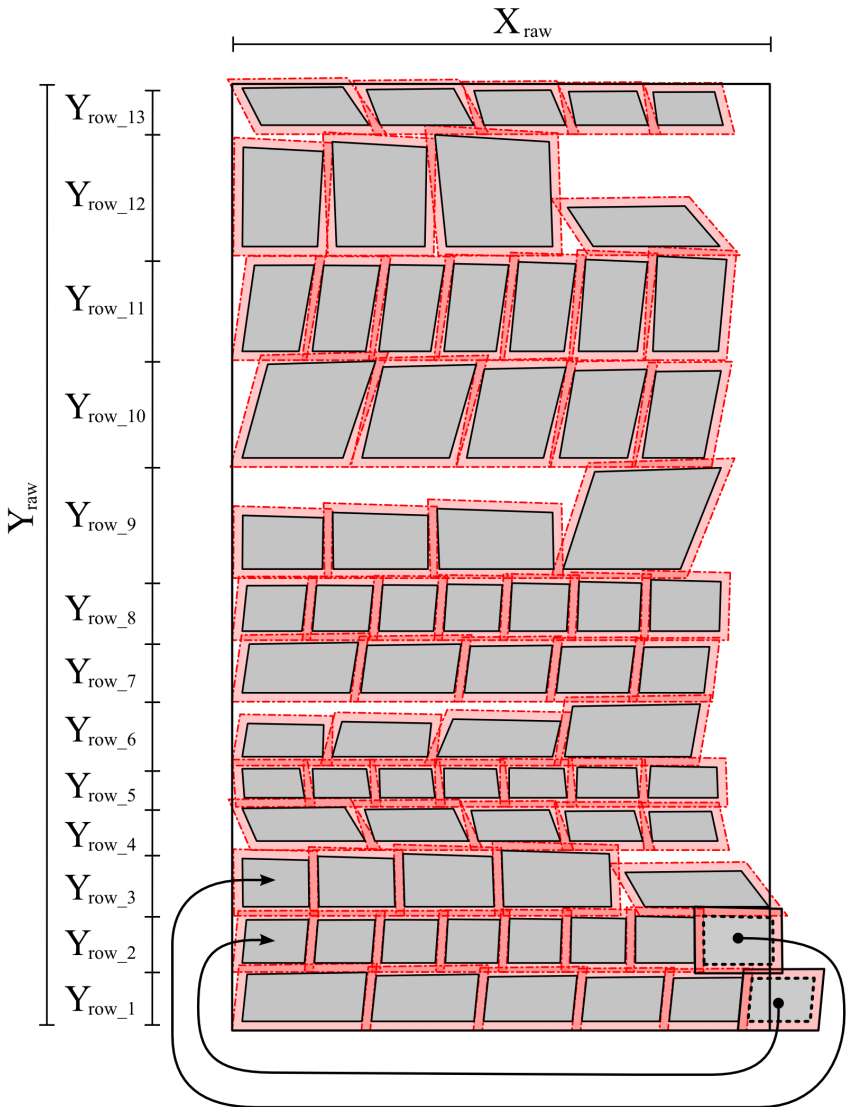


FIGURE 7.2.3. Packing one complete container

Figure 7.2.3 shows one completely packed container. It seems obvious, that the described packing method is not at its optimum in terms of compactness and material waste implied. Actually, packing by the presented method leads in this example to about 15% of unpacked space in

the container. Manual packing would allow reducing free space of a container down to 10% or even 5% of empty space. However, this method has to be considered as a trade off between the time needed to pack parts for multi-part manufacturing and the costs of the raw material to be machined.

7.3. From Geometry Data to Machine Instruction

7.3.1. Machining conditions. The constructional elements are going to be produced by means of integrated manufacturing. Therefore, a 5-axis milling machine will be employed. Even though CNC-machines functioning with 6, 7 or more axis exist in the field of timber constructions, the actual problem regarding the generation of machine instructions can fully be explained working with the example of a 5-axis machine.

Within the scope of this work, the used machine was a small MAKAMM 7s driven by a BWO800 controlling unit. The dimension of the work area is 1.5m by 3m by 1.2m (XYZ), which is one the small end of machines used in the field of carpentry. Therefore, the size of the parts produced was limited by the machine dimensions.

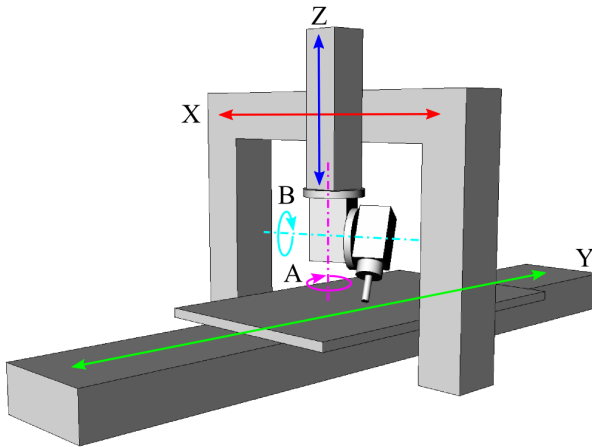


FIGURE 7.3.1. Maka MM 7s

Figure 7.3.1 shows a sketch of the used machine indicating the five work axis $\{X, Y, Z, A, B\}$, where $\{X, Y, Z\}$ are linear axis and $\{A, B\}$ rotational axis. The present machine has a fixed frame structure supporting the X- and Z-axis, while the Y-axis is implemented as a sliding working table. The fact that the machine has a working table will actually

condition the manufacturing methods. The machining strategies used for the production of the constructional elements have to take into account that the raw material could not be machined from below within the same manufacturing step (without turning and re-positioning the piece).

The machining instructions are based on the G-code standard (ISO 6983-1:1982), which describes the basic machine commands such as rapid moves or controlled feed moves in a straight lines or arcs. Further, it allows to operate tool changes, control spindle speed, stopping the machine, etc., etc. However, the used controller unit BWO800 provides more or less specific instructions extending the ISO-standard to machine specific instructions. They allow controlling the vacuum table, the cooling system, the exhaust system, etc. Aiming to provide generic G-code able to work on most of the 5-axis machines, we try to limit the machine specific instructions to its strict minimum.

Basically, machine instructions contain the coordinates $\{X, Y, Z, A, B\}$, which normally define the position of the tool tip. A set of two successive coordinates may hereby determine a linear translation, which can be driven at a given speed. Moreover, depending on the material and the plunging depth, the spindle speed can be defined.

7.3.2. Machining strategies. The prepared mapped and packed geometry data of the constructional elements is translated into a set of machine instructions. Therefore, a set of machining strategies has been worked out. Basically, the machining strategies correspond to different machining work steps, as for example: Address engraving, detail machining, work piece fixation, cutting operations, etc. Each of these machining strategies has further been split into a set of sub-strategies depending on geometrical aspects of the work piece being cut and the tool used for the machining step.

Below, a set of three cutting strategies are presented in order to illustrate the process allowing the automated generation of machine instructions. The strategies compare three different ways of cutting the chamfered edges of quadrilateral timber panels, which are:

- Pocketing, cf. figure 7.3.2
- Contour cut using a circular saw blade, cf. figure 7.3.3
- Contour cut using a flat tipped finger tool, cf. figure 7.3.4

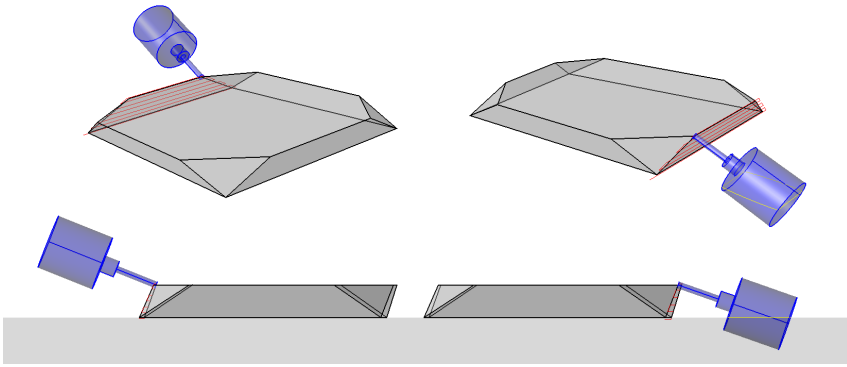


FIGURE 7.3.2. Pocketing

Machining pockets can be seen as a surfacing operation. Basically, a rectangle needs to be defined in which the controller surfaces off everything inside the given boundary. Surfacing can be done either by spiraling or by linearly sweeping. A flat tool tip will surface the plane being machined completely. Although this machining strategy is slower than the ones described below, it is highly necessary in case the two others are not applicable for geometrical reasons.

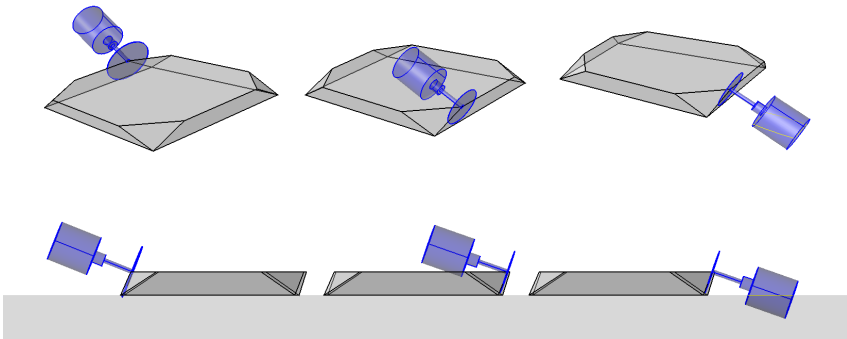


FIGURE 7.3.3. Contour cut using a circular saw blade

Machining straight cuts using a circular saw blade is providing the most proper parts in terms of finishing. The instructions provided to the machine are rather simple: 1. take the oriented position 2. plunge 3. cut a straight line 4. lift up. The instructions are highly resembling the ones used for contour cut machined with a flat tipped tool (see below). The initial orientation of the tool is though different, since the it is right-angled

compared to the flat tipped finger tool. One problem related to machining with circular disk blades is the free space needed for their operation. This is influencing on the distance from one part to the other. Ideally, circular saw blades are employed for machining parts being fixed on stands. This provides the machine more free space around the work piece and allows to work from below. Since the CNC-machine used here presents a working table, on which are fixed the parts, the actual free manoeuvring space is greatly restricted. Several machine/tool and machine/workpiece collisions are illustrated by figure 7.3.3.

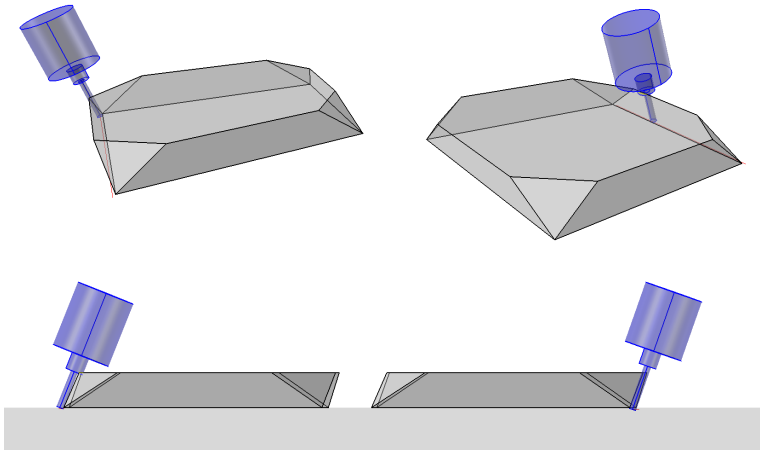


FIGURE 7.3.4. Contour cut using a flat tipped finger tool

Contour cut using flat tipped finger tools (cf. figure 7.3.4) will be the preferred machining strategy to use whenever possible. It is the fastest machining strategy providing the least material waste. Similar to the contour cut done by circular disc saws, the programming is simple. Since the tools used for machining are smaller in size than large circular saw blades, the machining path is shorter and the demanded free space to operate is sensibly smaller. This allows to pack the pieces to be produced in a more compact way, as illustrated in section 7.2.

7.3.3. Tool path generation. The tool paths are generated in function of the work piece geometry and the machining strategy chosen. Figure 7.3.5 shows a representation of the different tool paths used for the manufacturing of a constructional element of a quad mesh structure. The tool paths shown in 7.3.5 describe the work steps being executed in the following order:

- (1) Magenta, 4mm Drill: Machining of holes for fixing the part by screws
- (2) Cyan, 45° conical engraving tool: Address engraving
- (3) Green, 8mm Drill: Machining of holes for fasteners
- (4) Red, 20mm flat tipped finger tool: 3-pass contour cut

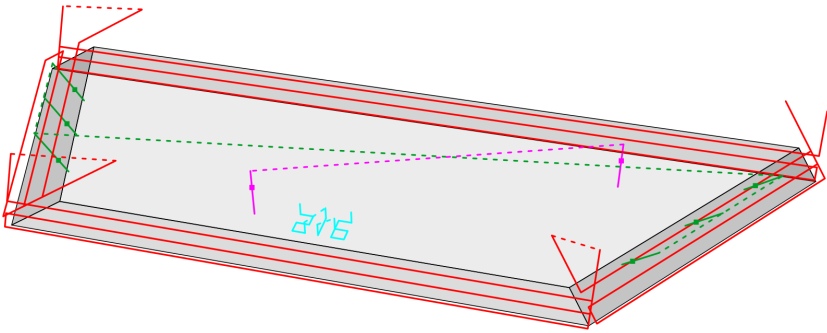


FIGURE 7.3.5. Tool paths for the manufacturing of one chamfered timber panel

Besides the color codes, which represent different machining tools used, the illustration of the tool paths shown in figure 7.3.5 distinguishes between rapid moves - G0-instructions executed at the maximum machine speed; dashed lines - and machining moves - G1-instructions executed at a given feed rate; continuous lines.

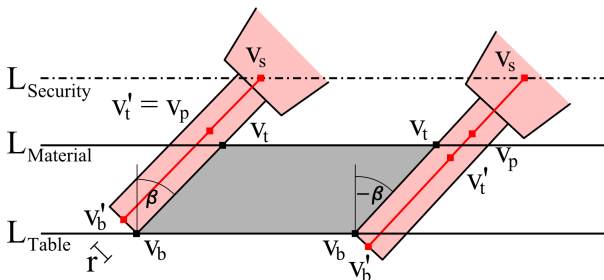


FIGURE 7.3.6. Section view of tool path offset generation

Figure 7.3.6 illustrates schematically how the contour cut tool paths may be generated. It shows two different cutting situations where the chamfer cut has either a positive angle $\{\beta\}$ or a negative angle $\{-\beta\}$.

To represent chamfer surfaces by containing rectangles allows to employ the same method for the tool path generation, independently from the chamfer surface's geometry. More generally, several cutting operations as well as some of the detail manufacturing operations (e.g. to groove the slot presented in figure 6.4.1) may be described by oriented containing rectangles.

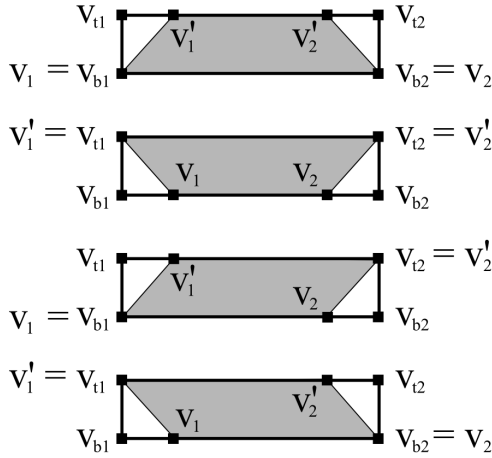


FIGURE 7.3.8. Rectangle containing the chamfer surface

Figure 7.3.7 shows the tool path parameters of one isolated chamfer cut in a axonometric view. The chamfer surface $\{v_1, v_1', v_2, v_2'\}$ is rendered in gray. The rectangle containing the chamfer surface is defined by $\{v_{b1}, v_{t1}, v_{t2}, v_{b2}\}$. There exist four possible relations between the containing rectangle and the chamfer surface, which are shown by figure 7.3.8.

7.3.4. From tool-paths to G-code. The geometry of the cutting tool path has been defined within section 7.3.3. The tool path geometry has to be translated into G-code instructions. Within the following list, generic pseudo G-code sequence of a chamfer cut (according to figure 7.3.7) will be explained point by point:

- (1) T10 M3 S18000 [Take tool number 10, start turning it clockwise at 18000 RPM]
- (2) G0 X(v'_{s1}) Y(v'_{s1}) Z(v'_{s1}) A(\vec{v}) B(\vec{v}) [Rapid move to point v'_{s1} , orient tool collinear to \vec{v}]
- (3) G1 X(v'_{t1}) Y(v'_{t1}) Z(v'_{t1}) F5000 [Move to point v'_{t1} at the approaching feed rate of 5000 mm/min.]
- (4) G1 X(v'_{b1}) Y(v'_{b1}) Z(v'_{b1}) F1500 [Move to point v'_{b1} at the plunging feed rate of 1500 mm/min.]
- (5) G1 X(v'_{b2}) Y(v'_{b2}) Z(v'_{b2}) F3000 [Move to point v'_{b2} at the cutting feed rate of 3000 mm/min.]
- (6) G1 X(v'_{t2}) Y(v'_{t2}) Z(v'_{t2}) F4000 [Move to point v'_{t2} at the lifting feed rate of 4000 mm/min.]
- (7) G1 X(v'_{s2}) Y(v'_{s2}) Z(v'_{s2}) F5000 [Move to point v'_{s2} at the approaching feed rate of 5000 mm/min.]

Point (1) operates a tool change. This is actually not strictly related to the execution of one specific chamfer cut. However, point (1) needs to be executed before the first chamfer cut is going to be machined. Within the following G-code lines {(2)...(7)}, the variables in brackets, e.g. (v'_{s1}), represent the respective coordinate values of the vertices $\{v_i\}$ defining the tool-path polygon. In addition to the $\{X, Y, Z\}$ -coordinates, command line (2) orients the tool by means of $\{A, B\}$ -coordinates. These coordinates contain angular values expressed in degrees. The tool's orientation within the machine space is defined the vector \vec{v} . The direction vector \vec{v} is obtained in function of the tool path vertices $\{v'_{b1}, v'_{t1}\}$ and can be computed by following expression: $\vec{v} = \overrightarrow{v'_{b1}v'_{t1}}$.

Figure 7.3.9 illustrates the components $\{v_x, v_y, v_z\}$ of the direction vector \vec{v} . They are used to compute the axis coordinates $\{A, B\}$. Note: According to the machine settings, $A = 0$ is defined as being collinear to the Y-axis, while $B = 0$ is defined being collinear to the Z-axis.

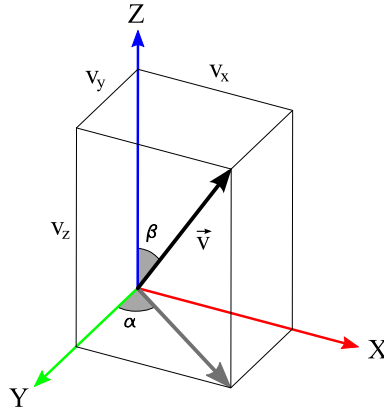


FIGURE 7.3.9. $\{A, B\}$ -angles from vector

According to this, the value of the A-axis angle α is defined by:

$$\alpha = \arcsin \left(\frac{v_x}{\sqrt{v_x^2 + v_y^2}} \right)$$

The value of the B-axis angle β is defined by:

$$\beta = \arccos(v_z)$$

Since the trigonometric functions are periodic, the sign of the angles need to be verified, in order to define $\{A, B\}$.

- If $v_x < 0$, then $A = -\alpha$ else $A = \alpha$
- If $v_y < 0$, then $B = -\beta$ else $B = \beta$

Nota bene: The relation established between \vec{v} and $\{A, B\}$ is actually depending on the type of CNC-machine used. The geometry of the A- and B-axis of the used MAKKA MM 7s is right angled (cf. figure 7.3.1). However, other CNC-machines may have different implementations of the two rotational axis.

7.4. Testing and discussion

7.4.1. Sample production. To test the established digital chain, several timber elements have been produced. The verification of the element's geometries has been done by the production of MDF quad elements. Figure 7.4.1 shows the result of the first production tests. The used tool-paths for the production of the first pieces have been obtained

by the methods discussed above. They are illustrated by figure 7.3.5. The production time per part was between one and two minutes (without counting the time necessary to mount and dismount the machine). This can be considered as highly efficient.

Actually, the list of machining strategies has been optimized with regards to the constructional elements. The fabrications of different constructional elements will presumably need to extend the developed machining strategies in order to provide equally efficient results. More examples will be shown by a series of specific applications within the following chapter.

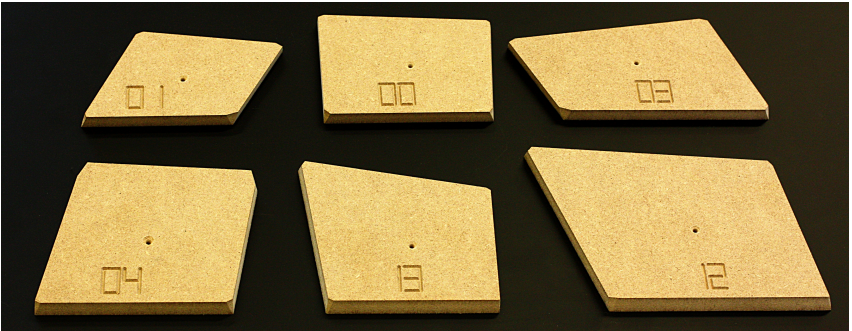


FIGURE 7.4.1. CNC-machined MDF quad elements

The actual production tests have been carried out on a MAKAMM 7s 5-axis CNC-milling machine. Even though the production samples have been machined on a specific device, the general production strategies worked out in this chapter can be adapted to other CNC-machines of different manufacturers. This adaption would need to adjust the G-code post-processor according to the concerned controller and the machines geometry. As presented in the appendix (cf. 11.3.3) the post-processor has already successfully been ported to two other CNC-routers. As long as the used manufacturing devices can be driven by a standardized set of instructions, the methods developed within the scope of this chapter may be applied without any broader problems.

The proposed mapping and packing method works flawlessly and has been applied on several prototypes, which will be presented in the next chapter. Its advantage is that it is fairly rapid compared to manual packing and a lot of 2D-packing solutions existing in the market. Furthermore, it provides the possibility to offer a packing method for multi-part 5-axis

manufacturing which actually does not exist in the industry. It is optimized to specific constructional parts and does integrate parameters from the level of machining (tool geometry, machining strategy, etc.). On the one hand, it does not conduct to an absolute optimum packing solution in terms of waste/raw material ratio. On the other hand, it provides a method for automated mapping and packing allowing to save costly labor time of the operator.

CHAPTER 8

Applications by Examples

The series of examples shown within the scope of this chapter illustrate the potential of iterative geometric design for architectural use. The applications range from purely decorative and formal applications to structural and load bearing applications. The shown cases treat equally questions related to the geometric design and their production. Depending on the type and scale of application different design and production techniques are implied.

8.1. Decor Panels

Two examples shall be mentioned for demonstrating the design capabilities of iterative geometric design used for fractal modeling with regards to applications of formal and ornamental character. The shown applications are basically designed by two dimensional fractal curves. The production process of the prototypes are not explained in detail, since the fabrication of two dimensional curves by means of 3-axis CNC-machines does not present any inherent difficulties. The discrete fractal curves are used literally as tool paths employed for machining.

8.1.1. Hilbert Panels. Application number one is about fractal tiling and ornamental decorative panels. This design of the fractal curve is obtained by a variation of the Hilbert curve, defined by the German mathematician David Hilbert in [Hil91]. The curve is defined by four affine geometric transformations, combining scaling, rotations, symmetries and translations.

The construction principle of the curve is explained by figure 8.1.1. The top row illustrates the four transformations $\{T_1, T_2, T_3, T_4\}$ while the bottom row shows the first two construction steps of the curve. Note, that the germ, the initial geometric K object on which the transformations are going to be applied, is a u-shaped polygon in the present example.

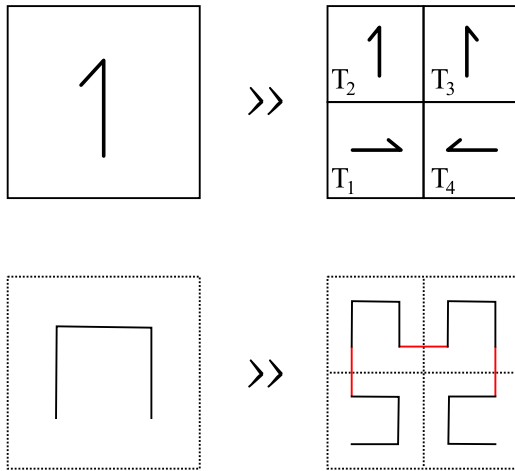


FIGURE 8.1.1. Construction principle of the Hilbert curve

The design of a Hilbert curve converges to a completely space filling curve object. However, the intermediate construction states show interesting patterns, which actually depend on the germ K used for the construction of the curve. Figure 8.1.2 shows the parameters used to obtain the variations of the Hilbert curve: the control point polygon $\{p_i\}$ and the germ polygon $\{k_i\}$

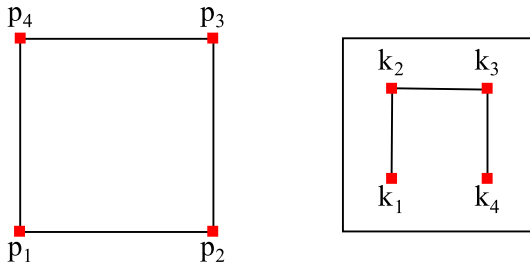


FIGURE 8.1.2. Hilbert curve parameters

The parameters define a control point polygon $\{p_i\}$ which allows to deform and manipulate the shape of the curve globally. The germ polygon $\{k_i\}$ defines the shape of the initial object on which the transformations are going to be applied. By modifying the size and the shape of the germ, interesting patterns emerge at the intermediate construction steps

of the curve. Figure 8.1.3 shows some possible Hilbert curve variations obtained only by changes applied to the germ.

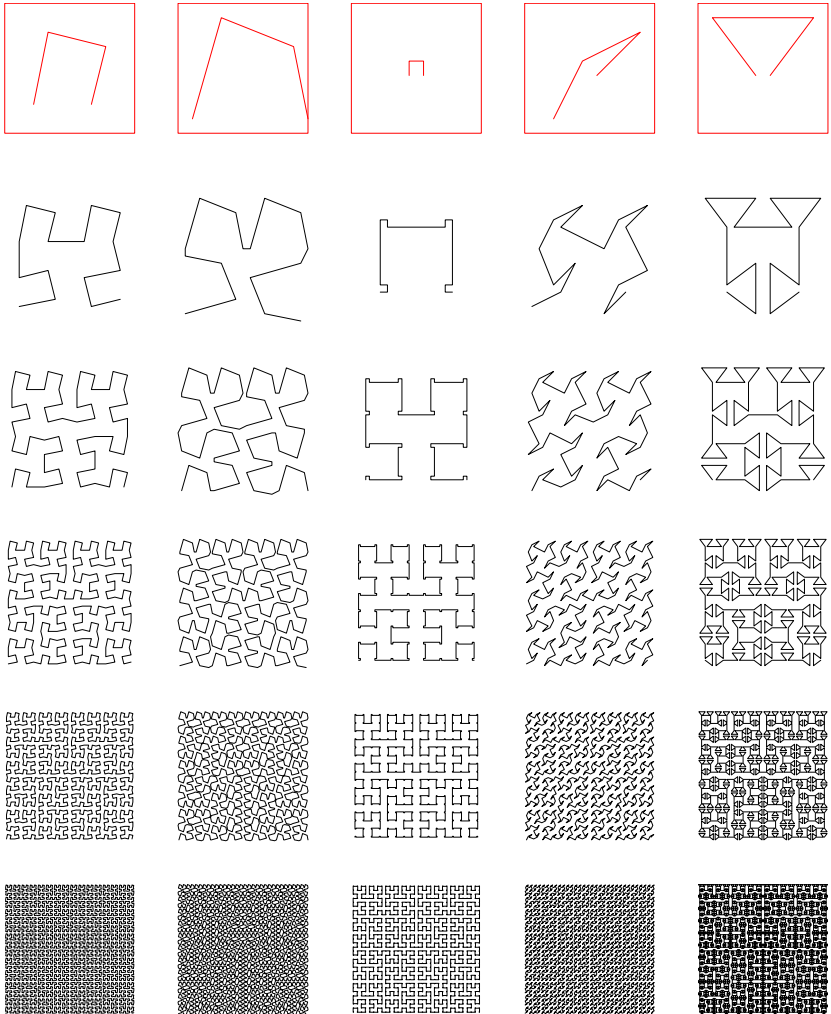


FIGURE 8.1.3. Hilbert curve variations

The following figures {8.1.4, 8.1.5} show 10 different decor panels which have variations of the Hilbert curve engraved. Note that the curves used for the panel's engraving are composed out of several patches of the same curve at different levels of iteration.

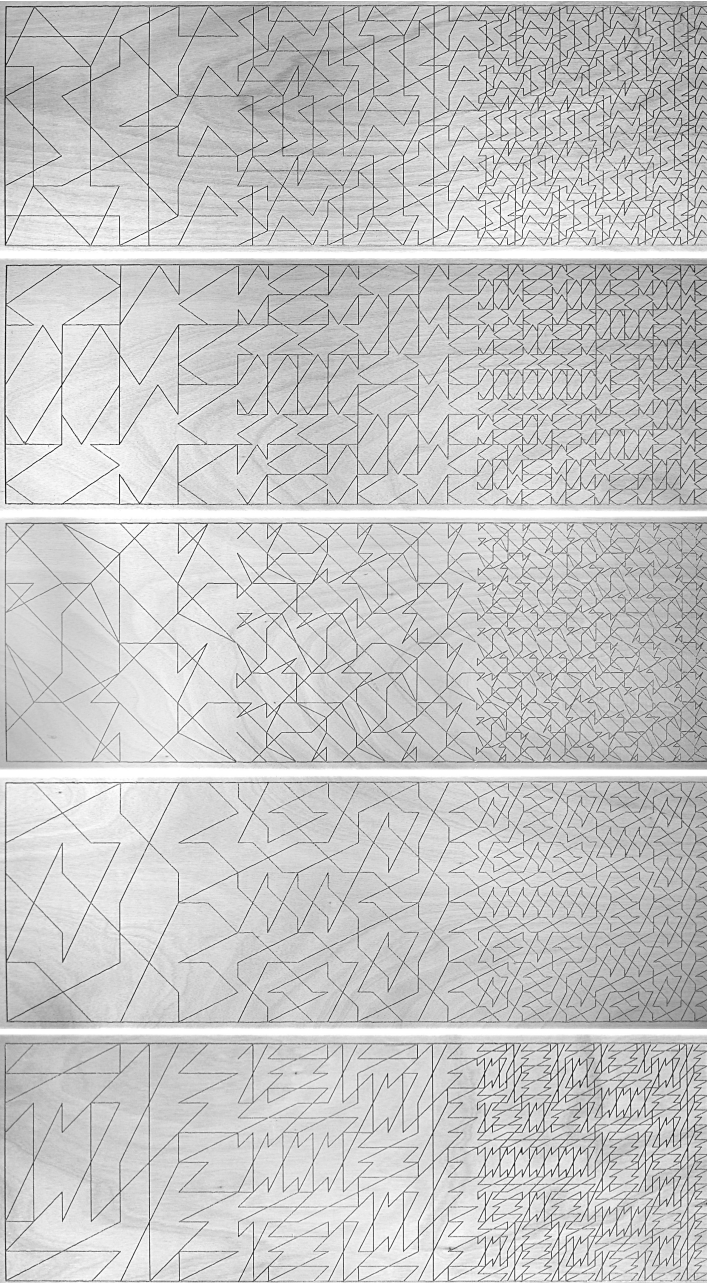


FIGURE 8.1.4. Hilbert tilings 1

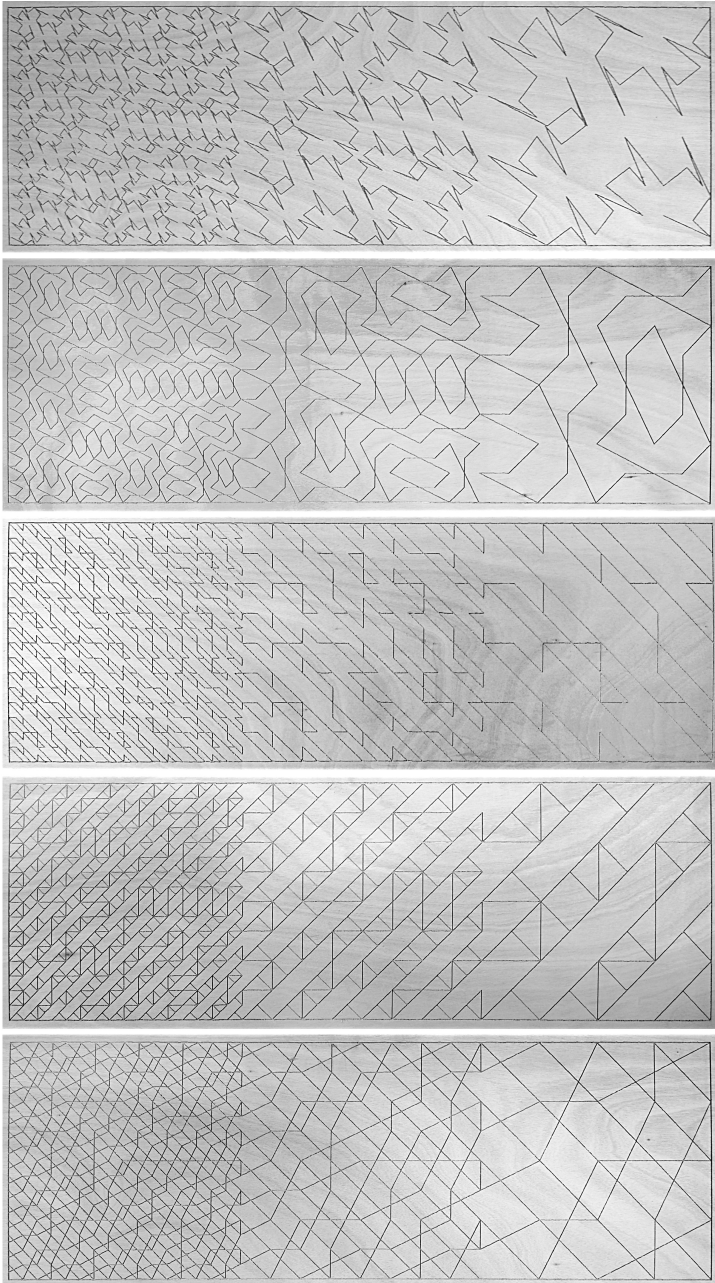


FIGURE 8.1.5. Hilbert tilings 2

8.1.2. Fractal Shading Panels. The design idea of the shading panels is inspired by the Dürer's pentagon described in [Dür77]. The fractal is based on a pentagon, which results into a snowflake like figure (cf. figure 8.1.6). Therefore it is sometimes referred to as pentaflake.

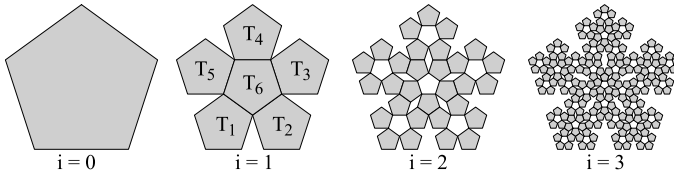


FIGURE 8.1.6. Dürer's Pentagon

The design of the shading panels uses an IFS-version of the Dürer's pentagon. The complete definition of the IFS used is described in [TBSG⁺06]. The design principle with the associated control and subdivision points are illustrated schematically by figure 8.1.7. The control handles provide an interactive way to act on the geometry of the pentaflake.

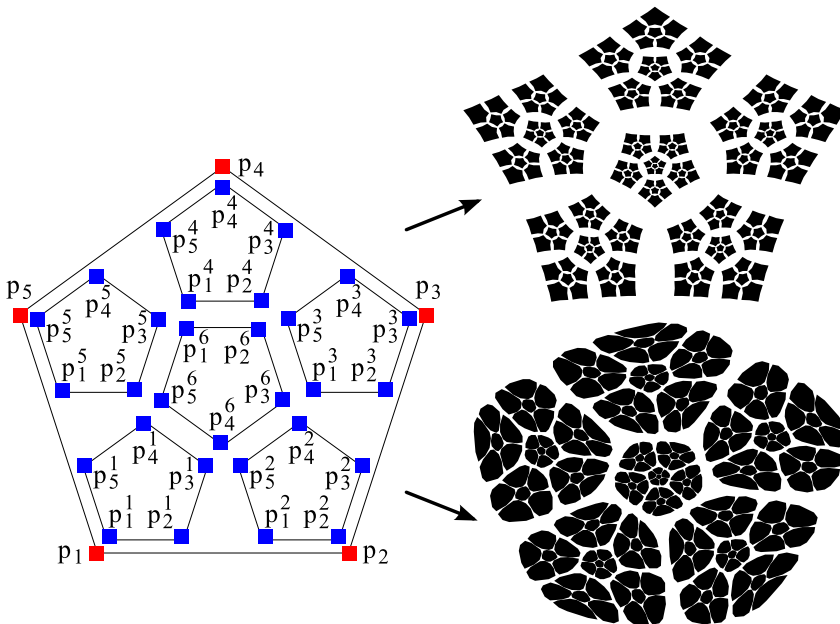


FIGURE 8.1.7. Parametrized pentaflake

The resulting figures have been machined out of thin plywood boards. The prototypes have been installed in our laboratory for a one year period. The presented examples of shading panels remind the design of arabesque window and privacy shades.

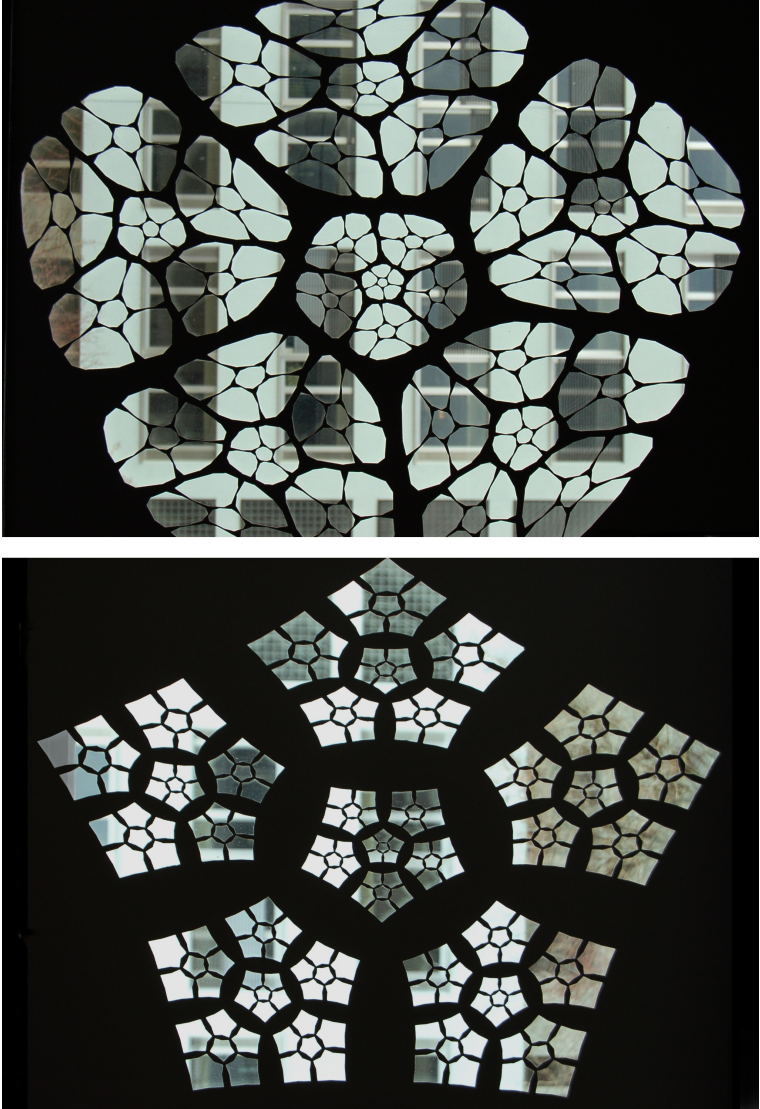


FIGURE 8.1.8. Two prototypes of fractal shading panels

8.1.3. Comments on ornamental applications. The examples presented above show literal applications of fractal iterative modeling to architectural elements. The designed geometric figures have been directly applied onto existing elements. This process is obviously quite straight forward, which means that there is no further development between the design and its application; and therefore may be considered a little oversimplified. However, the results deliver playful and apparently complex ornamental patterns that can be changed rapidly within the set of defined parameters.

8.2. Bézier Vault

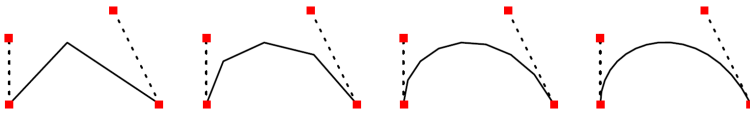


FIGURE 8.2.1. Iterative Bézier Curve

In this example, we build a vault structure based on an iteratively constructed Bézier curve with four control points (cf. figure 8.2.1). The straight line sections which build up the curve will be replaced by raw sawn timber planks. The planks section's dimension is 150mm by 35mm. The vault is composed of a series of curves which are put one next to the other. The spacing of the curves in the virtual 3D-model is 35mm, which corresponds to the thickness of the raw sawn timber planks. The planks are then screwed together in order to compose a massive timber vault structure. The shape of the vault's section can be controlled via the control points of the Bézier curve. Figure 8.2.2 shows a shape study where the curve's control points have been deformed such that the resulting shape is a meandering element with inflection points. The line segments of the underlying discrete Bézier curve have been replaced by constructional elements.

Once the shape has been defined, the curve will be subdivided into its parts until we obtain adequate lengths for the constructional elements. On the one hand, the lengths of the elements should not be longer than the prevalent planks existing on the market. On the other hand, the subdivision should be fine enough to obtain a smooth rendering of the curve. The overall design process is limited to two steps:

- Shape control, via the control points
- Subdivision control by choosing the adequate level of iteration

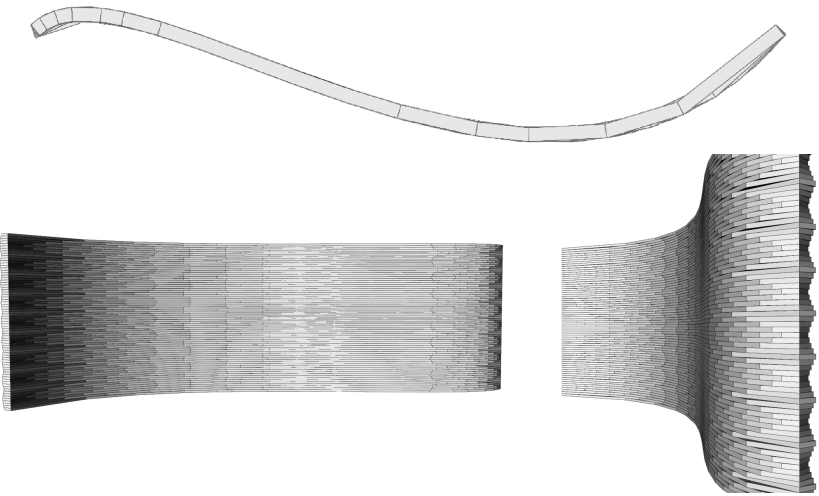


FIGURE 8.2.2. Shape study

8.2.1. Generation of the manufacturing data. The relevant dimensions, which are necessary for the production of the constructional elements, are directly induced by the geometric figure. The lengths of the planks correspond to the lengths of the curve's line sections. The chamfer angle can also be deduced from the geometric model (bisector angle of two adjacent line segments).

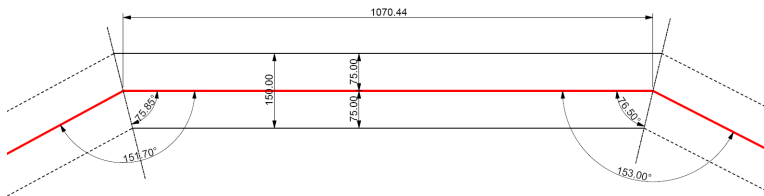


FIGURE 8.2.3. Relation between the geometry data of the discrete Bézier curve (shown in red) and the workshop drawing of one constructional element

The process described above, is illustrated figure 8.2.3. The line segment of the discrete Bézier curve lies in the middle. The length of this line segment defines the length of the constructional element. The offset curves, which define the width of the planks is just added in figure 8.2.3

as graphical representation, since the width of the actual construction element is already given by the raw material ($w = 150\text{mm}$). The entire manufacturing data is given by the angles $\{\alpha_{start}, \alpha_{end}\}$ and the length $\{l\}$.

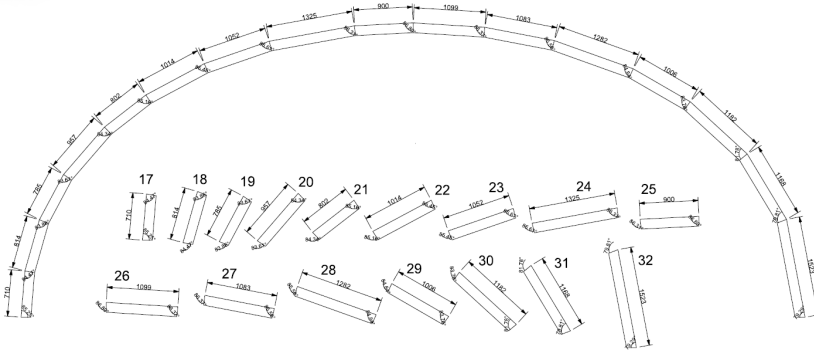


FIGURE 8.2.4. Workshop plans of one layer of the Bézier vault

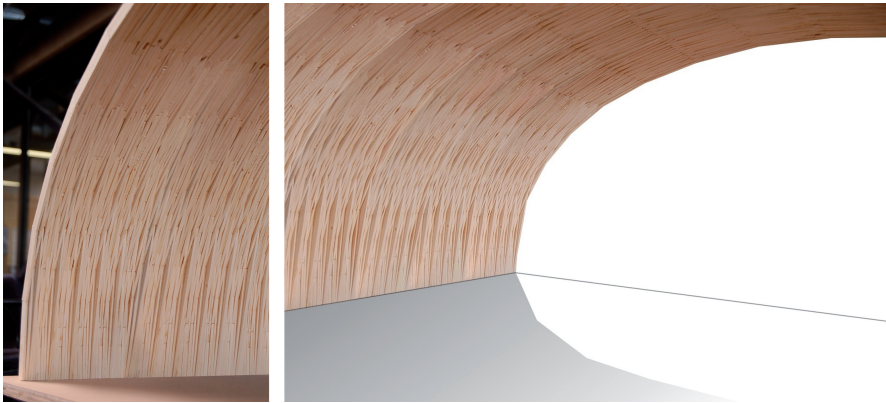


FIGURE 8.2.5. Reduced Scale Model

The set of manufacturing data per element may be represented by $\{\alpha_{start}, l, \alpha_{end}\}$. If we compute this data for each line segment of a cross section curve, we will end up with the complete set of execution plans for one layer (cf. figure 8.2.4). The numbering of the components use the addressing order provided by the IFS-formalism as presented in section 6.2. As stated above, the massive timber vault structure is defined by a

series of curves. Repeat this process for each curve and you will end up with the manufacturing data of every single constructional element of the entire building.

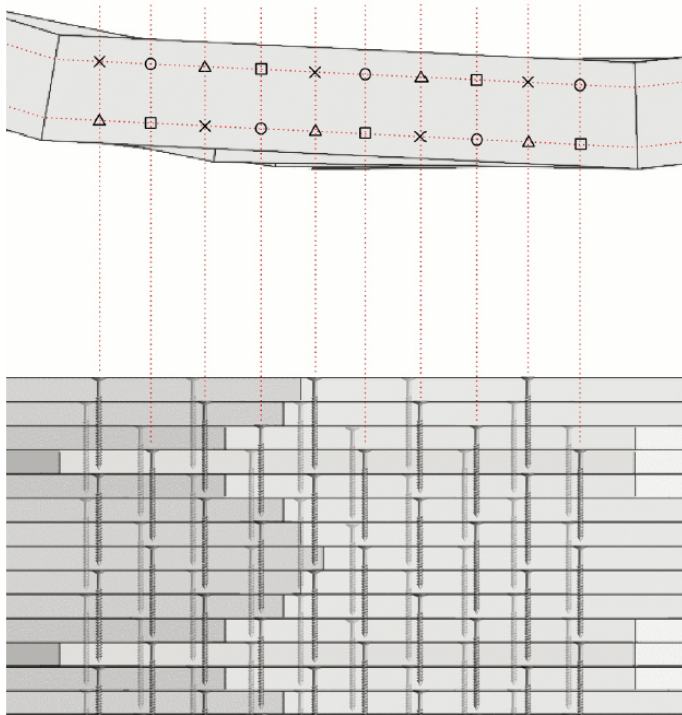


FIGURE 8.2.6. Construction Detail of the Bézier Vault Structure (top and front view)

Figure 8.2.5 shows a reduced scale model of such a massive timber vault structure. The manufacturing data has been obtained by the process explained above. The different planks have been cut by a computer numerically controlled routing table.

The question of how to partition a free-form object into a coherent set of constructional elements becomes obsolete because it is directly given by the iterative geometrical construction method. A direct link from design to production plans has been established, which is an important cost and time factor for the production of free-form architecture.

8.2.2. Note. Finally we would like to address briefly an important point which is important for detailing massive screwed plank structures.

In order to guarantee a minimum stiffness of the final object, we need to verify that the planks from one layer to the other are overlapping. Therefore, we need to verify the overlap during the design phase of the Bézier curves which build up the vault structure. As shown by figure 8.2.6 in an exemplary way, the screws link several layers of timber planks.

Therefore, the parameters defining the discrete Bézier curve have been carefully chosen in order to respect a minimal overlapping from one curve to the other.

8.3. B-spline Shell

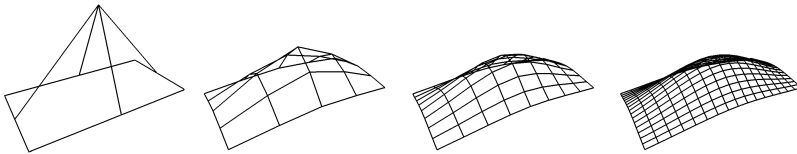


FIGURE 8.3.1. Four construction steps of a B-Spline-surface

The following example utilizes tensor products for the geometric design of IFS-surfaces. The used design method has been presented in section 5.4. In figure 8.3.1, the geometrically modeled surfaces are entirely composed of quadrilateral elements, called faces or quads. These faces, which are situated in the three dimensional space, are defined by four points called vertices. Generally, four points are not part of one common plane, which means that the modeled faces are not constrained to be coplanar. The construction material that we use for the physical construction is a planar timber panel. This puts up the following two questions:

- How to build using non-planar elements
- How to improve planarity of non-planar geometries

Ideally, it would be preferable to model directly geometric objects, which are entirely composed of planar faces. This is actually discussed in section 4.3 and implies the definition of additional geometric constraints which guarantee and verify the planarity of each face.

8.3.1. Bending planar construction material onto non-planar geometries. We propose to unroll each face in order to get a planar cutting pattern, which then will be automatically manufactured using CNC-machines. The produced pieces will then be constrained (bent) in order to reproduce the non-planar geometry of the initial faces. The limits of this

bending procedure are directly linked to the material properties of the employed timber panel. In order to limit the initial stress due to the bending of the wooden panel, minimizing the bending curvature is of great interest. Exceeding the limits of the material properties will result in fractured and therefore unusable construction panels. In order to minimize the curvature of the faces, a perturbation method is employed. This method acts on the IFS-Figure with the goal of reducing the local curvature of each face. The perturbation method will be discussed later on.

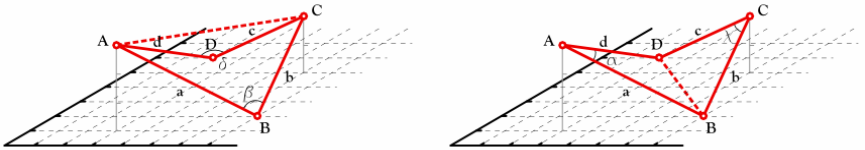


FIGURE 8.3.2. Non planar quads present two ways of triangulation. Depending on the angle value between the triangular planes the case presenting the lower curvature is chosen. This allows lessening the initial stress of the bent timber board.

Two ways to unroll a non-planar face are considered. Each of them is triangulating the quad into two triangles (cf. figure 8.3.2). The first triangulation gives us the triangles $\{A, B, C\}$ and $\{A, C, D\}$ while the second one gives the triangles $\{A, B, D\}$ and $\{B, C, D\}$. Assuming that each triangle forms a plane having a normal vector $\{\vec{n}_1, \vec{n}_2\}$, we are able to measure the angle between the two normal vectors by means of their dot product $\vec{n}_1 \bullet \vec{n}_2$. Note: The two normal vectors $\{\vec{n}_1, \vec{n}_2\}$ is obtained by the unitized vector cross products of respectively $\|\vec{BC} \times \vec{BA}\|$ and $\|\vec{DA} \times \vec{DC}\|$. In general, the angle between the planes $\{A, B, D\}$ and $\{B, C, D\}$ are different from the angle between the planes $\{A, B, C\}$ and $\{A, C, D\}$. This property allows choosing the way of triangulation in function of the curvature value. Aiming to reduce the initial stress of the timber panels, the smaller angle between the surfaces normal vectors will be preferred.

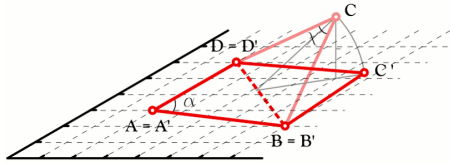


FIGURE 8.3.3. Quadrilateral faces are generally non planar. To unroll the shown quad $\{A, B, C, D\}$, point C is transformed by a rotation along the axis $[B, D]$, in order to be part of the plane $\{A, B, D\}$.

After having decided about the sense of triangulation, the non planar quad face is ready to be unrolled. Three points $\{A, B, D\}$ will be considered as fix points, which define the reference plane. As $\{A, B, C, D\}$ form a non planar quadrilateral face, point C is not part of the reference plane $\{A, B, D\}$. The transformation which will bring point C into the reference plane is a rotation along the axis $[B, D]$ (cf. figure 8.3.3). This way, the lengths of the four sides of the quad faces will remain unchanged. The only geometric data, which is actually affected, are the two angles next to point B respectively point D . Choosing the rotation rather than an orthogonal projection, has the advantage to conserve the length of the sides. This is highly important for the later assembly of the pieces, in order to guarantee perfectly closed joints in-between the quad elements. Finally we can state that we obtain a planar image $\{A', B', C', D'\}$ of the initial quad $\{A, B, C, D\}$ by one transformation on point C .

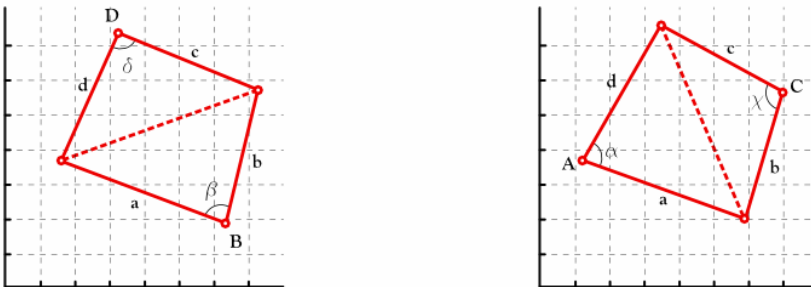


FIGURE 8.3.4. Unrolled cutting pattern: The Length of the sides and two opposite angles are preserved.

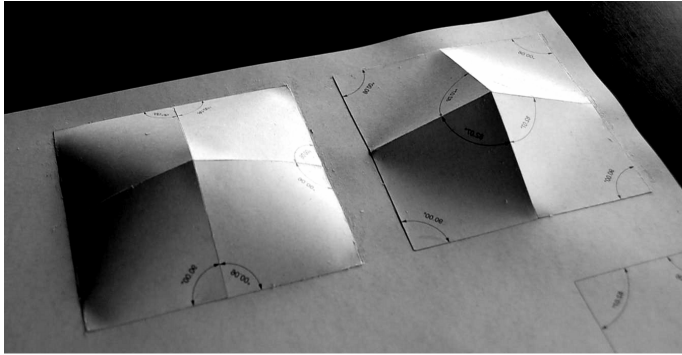


FIGURE 8.3.5. Non planar quads present two ways of triangulation. The rendering of the shape is a more or less accurate representation of the initially modeled geometric object.

Figure 8.3.5 illustrates a simple case of four non planar quad faces. The way the faces are triangulated does not depend on the criterion of the smallest angle, which is not always the most accurate way to represent the initial geometry. The different appearance of the two solutions shows the importance of the sense of triangulation. In the first case, the rendering of the object is much smoother than in the second case, where you can clearly identify the edges between the single faces. The sense of triangulation acts on the angles between the faces. The given example shows an extreme situation presenting a very important curvature angle. Further on, smaller curvature angles will be used for the physical construction out of timber panels.

8.3.2. Perturbation method to lessen the local curvature of non-planar quad meshes. The curvature analysis of an IFS-surface (cf. figure 8.3.6) shows that the greatest curvature is situated near the four corners of the initial geometric object, which is a cubic B-Spline surface. We propose a perturbation algorithm that acts on the vertices of the object tending to minimize the curvature of the elements constituting object. The presented method has been developed in collaboration with our project partner Iver Bailly-Salins.

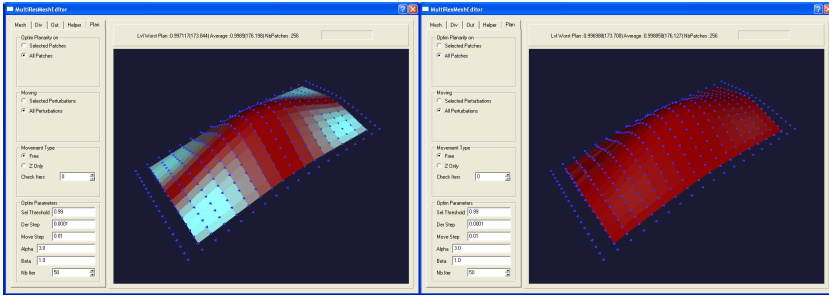


FIGURE 8.3.6. Tensor product surface of two cubic B-splines. [Left] Curvature analysis showing the greatest curvature near the corners of the shape. [Right] Geometric object after planarity optimization.

The planarization method makes an optimization of quad mesh faces. Each mesh is made of several faces, that are defined by four vertices. The goal of the planarity optimization is to get faces as planar as possible. The planarity p of each face is defined by the cosines of the angles between the corners. The closer it is to 1, the more the planarity is verified. A planarity is considered correct when this cosine is above a given threshold planarity p_t . According to the quad $\{A, B, C, D\}$ shown in figure 8.3.2, two planarity values $\{p_1, p_2\}$ can be computed according to the sens of triangulation :

$$p_1 = \frac{\overrightarrow{BC} \times \overrightarrow{BA}}{\|\overrightarrow{BC} \times \overrightarrow{BA}\|} \bullet \frac{\overrightarrow{DA} \times \overrightarrow{DC}}{\|\overrightarrow{DA} \times \overrightarrow{DC}\|} = \vec{n}_1 \bullet \vec{n}_2$$

$$p_2 = \frac{\overrightarrow{AB} \times \overrightarrow{AD}}{\|\overrightarrow{AB} \times \overrightarrow{AD}\|} \bullet \frac{\overrightarrow{CD} \times \overrightarrow{CB}}{\|\overrightarrow{CD} \times \overrightarrow{CB}\|} = \vec{n}_2 \bullet \vec{n}_1$$

The planarity optimization method we will consider p_1 , if $p_1 \leq p_2$, and with p_2 , if $p_1 > p_2$. If $p = 1$, then the planarity of the quad $\{A, B, C, D\}$ is verified, which means that $p_1 \parallel p_2$.

Within the following, the principle parameters that define the planarity optimization are presented:

- It is possible to act on selected areas of the figure, in order to complete the optimization method on a fragment of the mesh.
- The optimization of planarity can be applied partially on the mesh figure: I.e. only the faces less planar than a certain threshold will be optimized.
- The move step t is defining the initial step utilized to move points to improve planarity.

- Two coefficients, Alpha and Beta, define the importance respectively of the worst planarity and the average planarity in the optimization calculus. For example if Alpha is set to 1 and Beta to 2, then optimizing the average planarity is twice more important than optimizing the worst planarity.
- Finally, the number of iterations i made to improve planarity can be set.

To improve planarity by moving the points can last a very long time. The number of iteration lets one define in some way the time spent on the planarization. By increasing the number of iterations the calculation will take more time to complete but its result will tend to be more accurate in terms of planarity optimization.

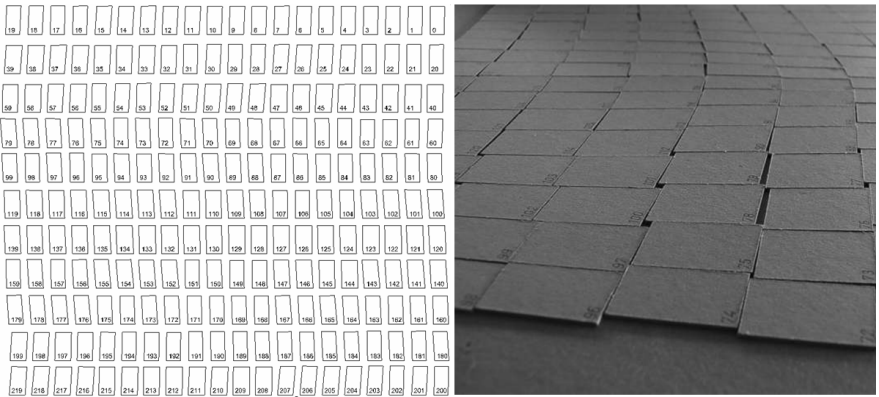


FIGURE 8.3.7. Each board, different in size and shape, receives a unique address. Automated generation of the CNC-files are prepared for integrated production and produced by a 3 axis cnc-machine.

8.3.3. Automation of the process. So far, the geometry of the cubic B-spline surface has been optimized. The quads that compose the surface have been planarized in order to minimize their local curvature values. Still, the quads are not completely planar yet. Therefore, each quad will have to be unrolled. This is done via a triangulation in function of the method explained in section 8.3.1. Once the quads are divided into two triangles, they are unrolled respectively unfolded and drawn on a 2D plane. This gives us the basis of the manufacturing plans. In addition to the geometric data, each flattened quad receives a unique address, since all the elements are different in shape and size. The addressing (unique

numbers) will allow the organization of the constructional elements in space, which is necessary for their assembly. The whole process of un-rolling, addressing, and NC-file creation has fully been automated.

The geometrical data of each board and its address are then written into machine code files for integrated production. As shown in figure 8.3.7 and 8.3.8, the method and the geometry are verified by the construction of a reduced scale card board model. The cut plates are first laid out on the ground according their address. The set of plates on the ground does not cover entirely the plane. There exist gaps in between the elements. The assembly of the elements has to assure a joint-less connection of the plates along their sides. This assembly was started in one corner, went through the whole B-Spline surface and ended in the opposite corner. If all the plates were assembled in a joint-less way, the assembly will automatically lead to the physical image of the initially modeled geometrical figure.



FIGURE 8.3.8. The method and the geometry are tested and verified on a reduced scale prototype

8.4. Timber Panel Structures

In this section we will discuss the application of iteratively constructed free-form surfaces to timber panel constructions. The surface method used for the design of the free-form objects has been described in section 4.3.

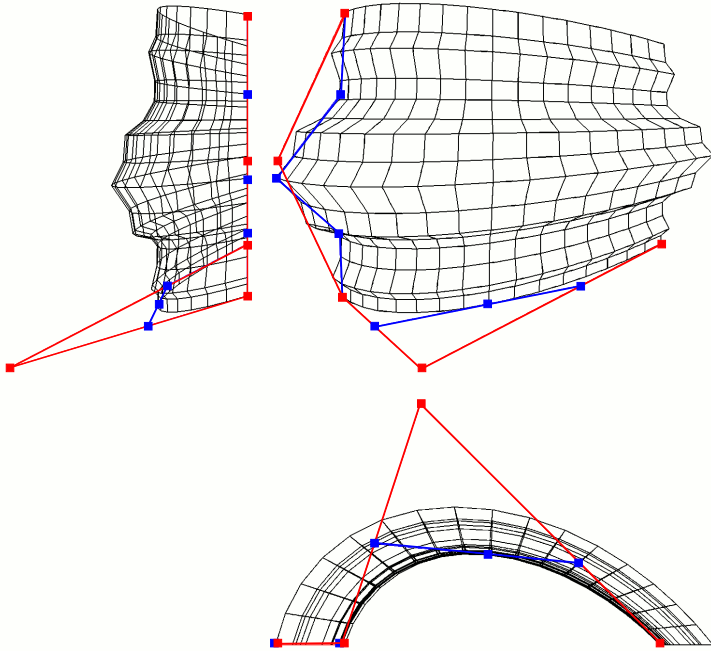


FIGURE 8.4.1. Top and side views of the surface design

Figure 8.4.1 shows a plan and side view of the a discrete surface describing the design of a prototype. The color rendition shows the control point polygon in red, while the subdivision point polygon is represented in blue. As described in section 4.3, the surface designed is completely defined by two edge curves. The present design study was used for the realization of the third prototype (cf. figure 8.4.9). It was mainly driven by following parameters: On the one hand, we wanted to realize a small dome-like structure, presenting a smooth arc in its longitudinal section. On the other hand, we designed a rough curve, providing folds to the transversal section of the structure. We will come back to this design later in this section.

Once the geometric figure has been designed, it will be translated into a set of constructional elements.

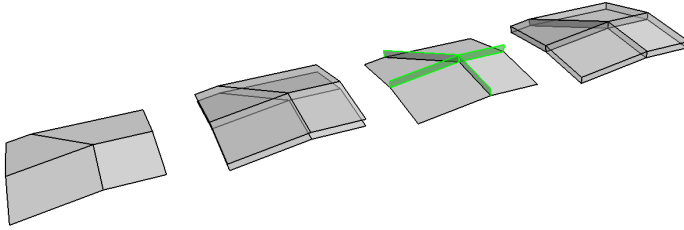


FIGURE 8.4.2. Parallel Offset Mesh Generation

In the present example, the faces that compose the discrete surface are replaced by planar timber panels. The choice of the thickness of the timber panel is important as the virtual 3D-surface does not present any thickness. A volume model has to be derived from the surface model. The thickening process is illustrated by figure 8.4.2. First, we generate a parallel offset surface, which holds a constant distance to the initial surface. The distance corresponds to the thickness of the timber panel. Second, the bisector planes are calculated, we will use them later for the chamfer cut of the panels. In this way, we design free-form objects that are entirely built up of planar constructional elements. Please consider section 6.3 for a more in-depth description of the employed thickening method. Figure 8.4.3 shows an example of an IFS-surface whose faces have been entirely replaced by thickened constructional elements.

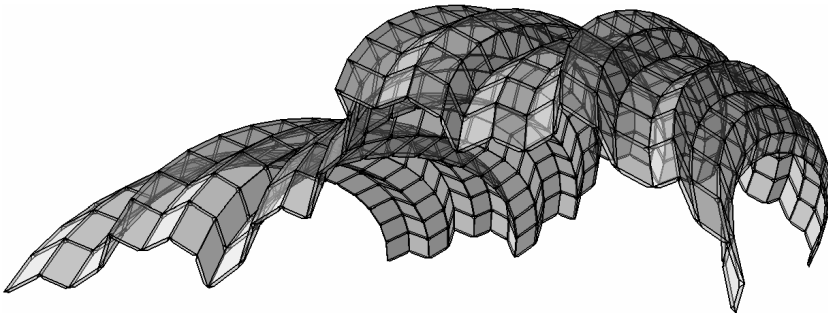


FIGURE 8.4.3. Thickened ISF-surface

8.4.1. Manufacturing. We will summarize briefly questions related to integrated production of the constructional elements for the timber plate structure. Please see chapter 7 for detailed information about the manufacturing process implied for the production of discrete geometric design. The procedure to get from the geometry data of the constructional elements to the machine code has been mainly automated.

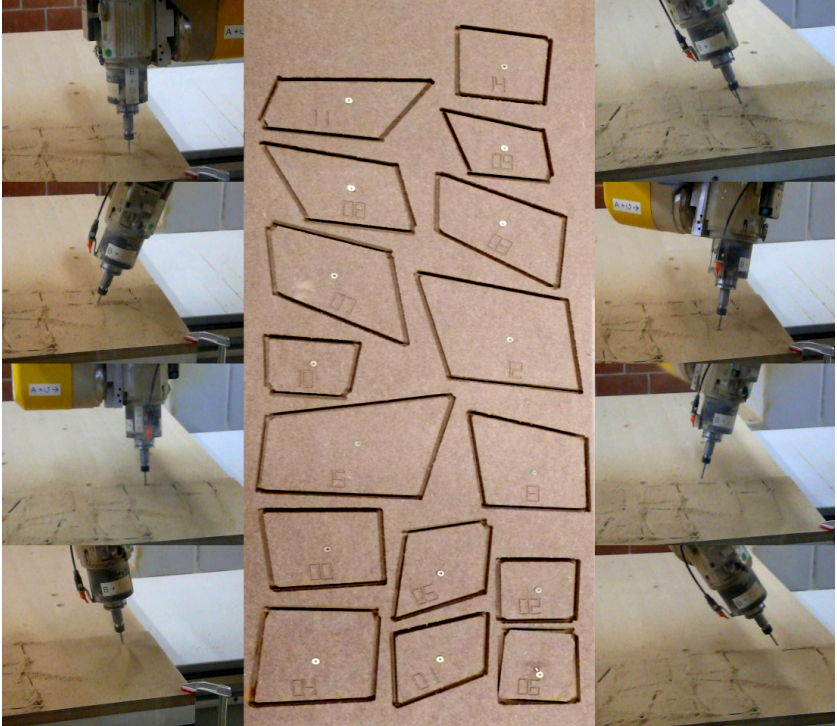


FIGURE 8.4.4. Integrated manufacturing of the constructional elements

To realize the parts of such complex shapes, the following work steps are necessary:

- A unique address for each constructional element is necessary for the logistical reason that the different elements can be assembled in the right place.
- Each element has to be oriented according to the coordinate system of the CNC-machine, the dimensions of the raw material and fiber direction of the plywood panel.

- Automatic generation of the machine code for each element: The material properties, the type of machine and the nature of the cutting tools are of the highest importance for integrated production of the elements, which are all different in size and shape.

Figure 8.4.4 shows a sequence of the machining process. The production of each multi part plate has been split into three work steps:

- Piercing of the fixation holes. Each part is screwed on the machining table.
- Engraving of the addresses of each element.
- Contour cut: machining of the actual element.

8.4.2. Prototypes. The established digital design and production chain was verified and tested by means of a series of reduced scale prototypes. First, a partial prototype was going to be machined presenting an extract of a bigger structure. The realization of a detail of a bigger structure was intended to deliver insight about the implemented accuracy of the developed methods. Main manufacturing problems have not shown up during the production of the first constructional elements. However, due to the geometry of the constructional elements and the required precision needed to obtain perfectly fitting parts, several smaller adjustments needed to be done on the CNC-machine side. For example, the fixation of the parts needed to be enforced for the production of the plywood parts. Furthermore, the working table of the CNC-machine had to be calibrated several times, in order to guarantee exact positioning of the raw material boards, since a minimal shift of the Z-0-level induces major dimensional changes of the machined parts.

Finally, the assembled manufactured elements give an accurate rendering of the surface designed on the computer screen. This shows that practical realization of iteratively constructed surfaces becomes possible. After the realization of an early partial prototype (cf. figure 8.4.5), we went on testing the method on more complex structures. First, we completed an eight by eight mesh surface, just a little bigger than the initial test prototype shown in figure 8.4.5.

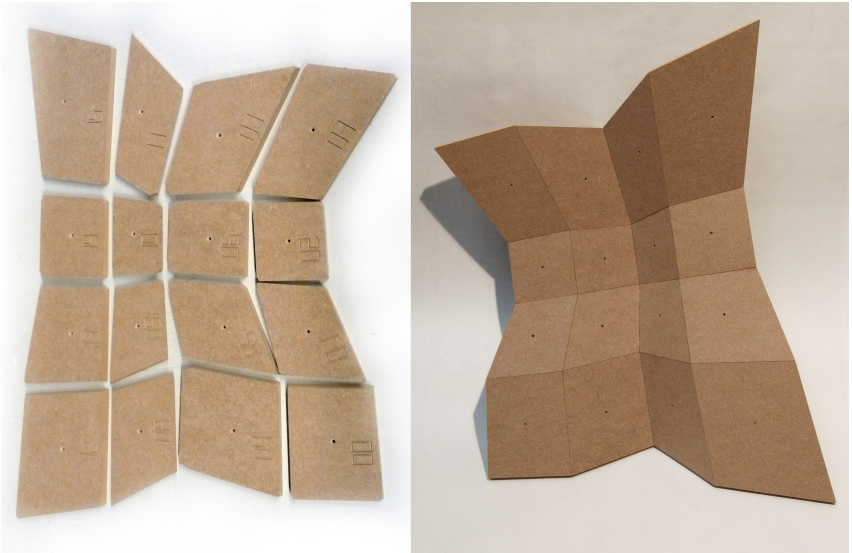


FIGURE 8.4.5. Assembling of the partial prototype

For the manufacturing of the second and the third prototype, mapping and packing strategies presented in chapter 7 have been applied. Figure 8.4.6 shows the mapped and packed parts of the second prototype - just after being machined.

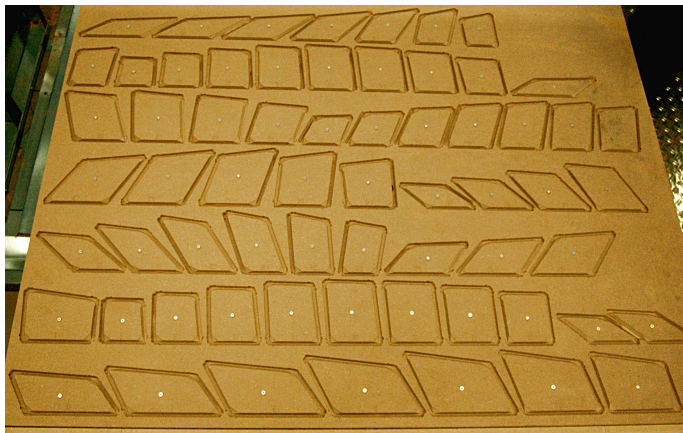


FIGURE 8.4.6. Machined elements of the second prototype

Figure 8.4.7 shows the layout of the mapped and packed parts of the third prototype. The black dots in the center of each part show the address numbers of the single pieces. Each, packed raw material board was then translated into one file of machine instructions for integrated production.

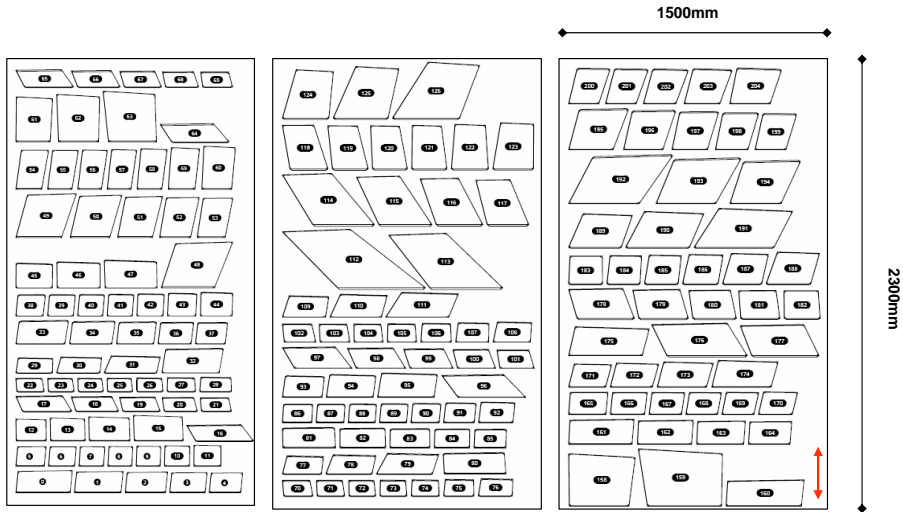


FIGURE 8.4.7. Workshop plans of the third prototype

Figure 8.4.8 shows the assembled second prototype composed of 64 elements. The photograph shows the front and the back side of the prototype. The engraved numbers on the back side of the prototype are the constructional element's IDs. The reason why the present prototypes present a different back side with regards to their front side is due to offset problems that the present free-form meshes provide, cf. section 6.3.3. Therefore, no consistent offset mesh was able to be computed and exceptions had to be taken in account for the vertices (the corner of the elements). We took advantage of the differentiation of the two sides to engrave the address IDs directly on the back side of the timber parts rather than sticking on paper labels containing the address information.

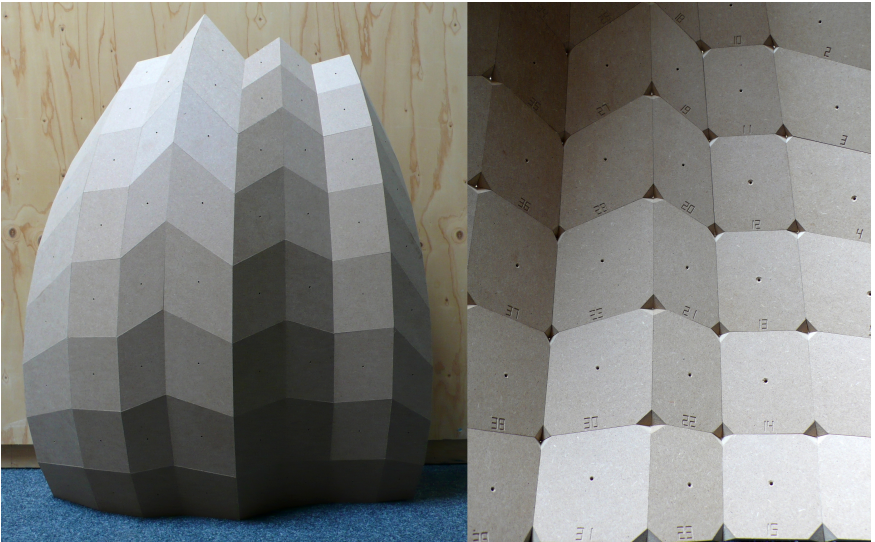


FIGURE 8.4.8. Second prototype - 8 by 8

After the realization of the first two small test surfaces, we approached the manufacturing of a bigger prototype composed of 256 constructional elements. The shell structure shown in figure 8.4.9 presents a small vault spanning over four and a half meters; its plates are 10 mm thick spruce plywood. The geometric design of this shell is shown in figure 8.4.1. It has been discussed in the beginning of this section. The production of the parts were done completely in an automated way. The manufacturing data included the generation of workshop plans shown in figure 8.4.7, showing three raw material panels of 10 mm thick plywood boards. The mapping of the elements to the raw material plywood panels is done with regards to wood grain direction, which is shown by the red double flash in graph number 8.4.7.

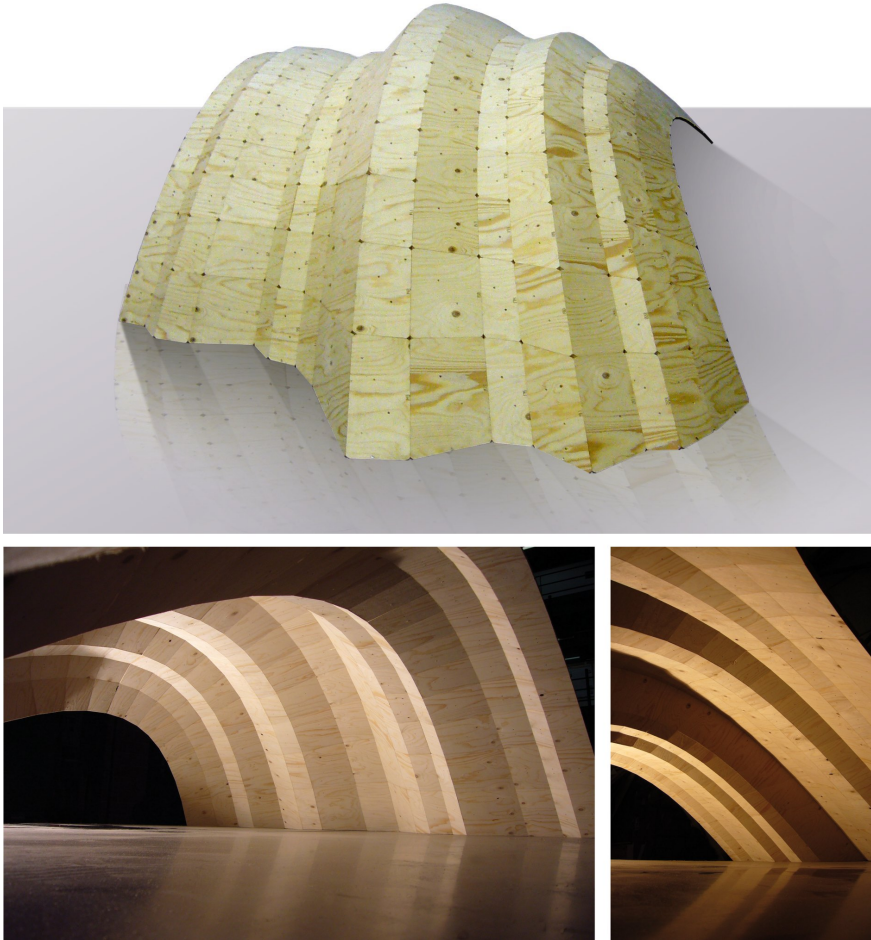


FIGURE 8.4.9. Interior and exterior views of the third prototype, which is constituted of 256 elements

The assembly of the parts was completed row-wise (cf. section 6.4.3), starting in the middle of the structure and ending at its sides. This way, we aimed to reduce the accumulation of relative precision errors due to manual assembly. The assembled prototype shows a nice little vault structure of which the parts fit astonishingly well. The shape of the ribs show a quasi-organic shape and integrate comfortably into the pleasing global dome structure.

8.4.3. Conclusions. The established methods for the design and the production of free form panel shells has shown to be quite robust. The

generation of the production data lasts only a few moments. Architectural design of the free-form surface aside, following production related properties need to be defined in order to obtain the complete set of production data:

- Material thickness (needed for the offset mesh generation according to chapter 6.3)
- Definition of front and back side of the surface
- Dimensions of the raw timber panels (needed for mapping and packing)
- Tool data: geometry, feed rates, rpm, etc. (needed for the generation of the G-code according to chapter 7)

Hereby, we end up with a general description of free-form surfaces built by planar timber elements. This versatility allows us to quickly change the initial design or let us use another raw material board. Possible upcoming changes in the design cycle can be integrated easily and quickly to the final product. Note: that the needed time for milling and assembly is actually not take into account. After the realization of the three presented prototypes we were testing a series of new designs at a larger scale. Withing the following pages, two design studies are shown. The two prototypes number four and number five present variations of the prototype number three. The scale of those designs has doubled and the number of elements has also be subject to changes. Further, the geometry of the surface has also be changed slightly in order to get a more elongated structure (in case of study #4) or a more square dome-like surface (in case of study #5). The material was thought to be 21mm thick three layered timber bloc panels.

The figures 8.4.10 to 8.4.13 show the two design studies and its corresponding workshop plans. Further focus is driven to the size of respectively the smallest and the biggest element of the structures designed. We see that this relation can heavily vary depending on a given design. Therefore, during the design phase, special attention should be payed to the size of the resulting elements, e.g. we should assure, that no part is actually being bigger than the size of the employed raw material.

Although the realized objects remain till today relatively small-sized, they allowed us to verify the validity of the proposed design method, since the employed manufacturing techniques are also applicable for real scale constructional elements.

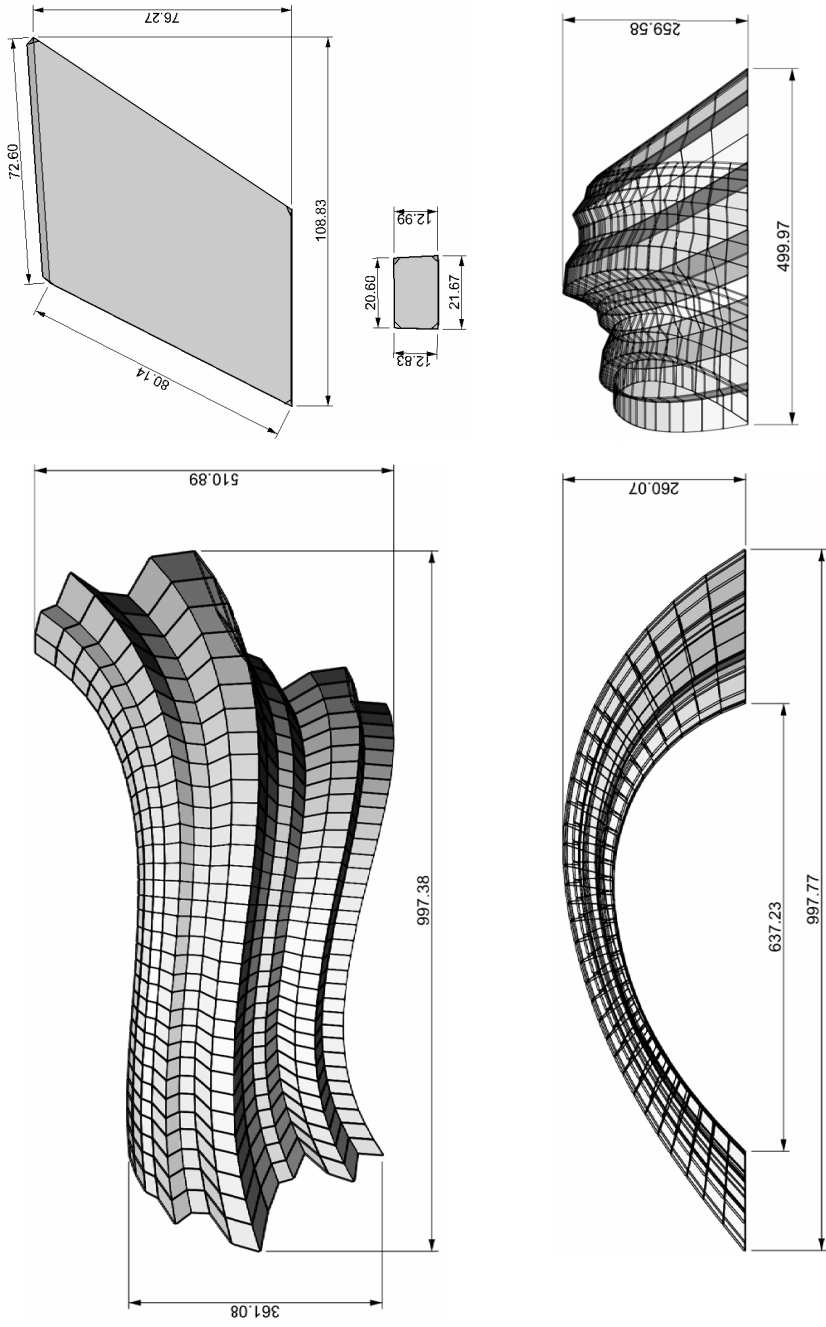


FIGURE 8.4.10. Design of prototype #4

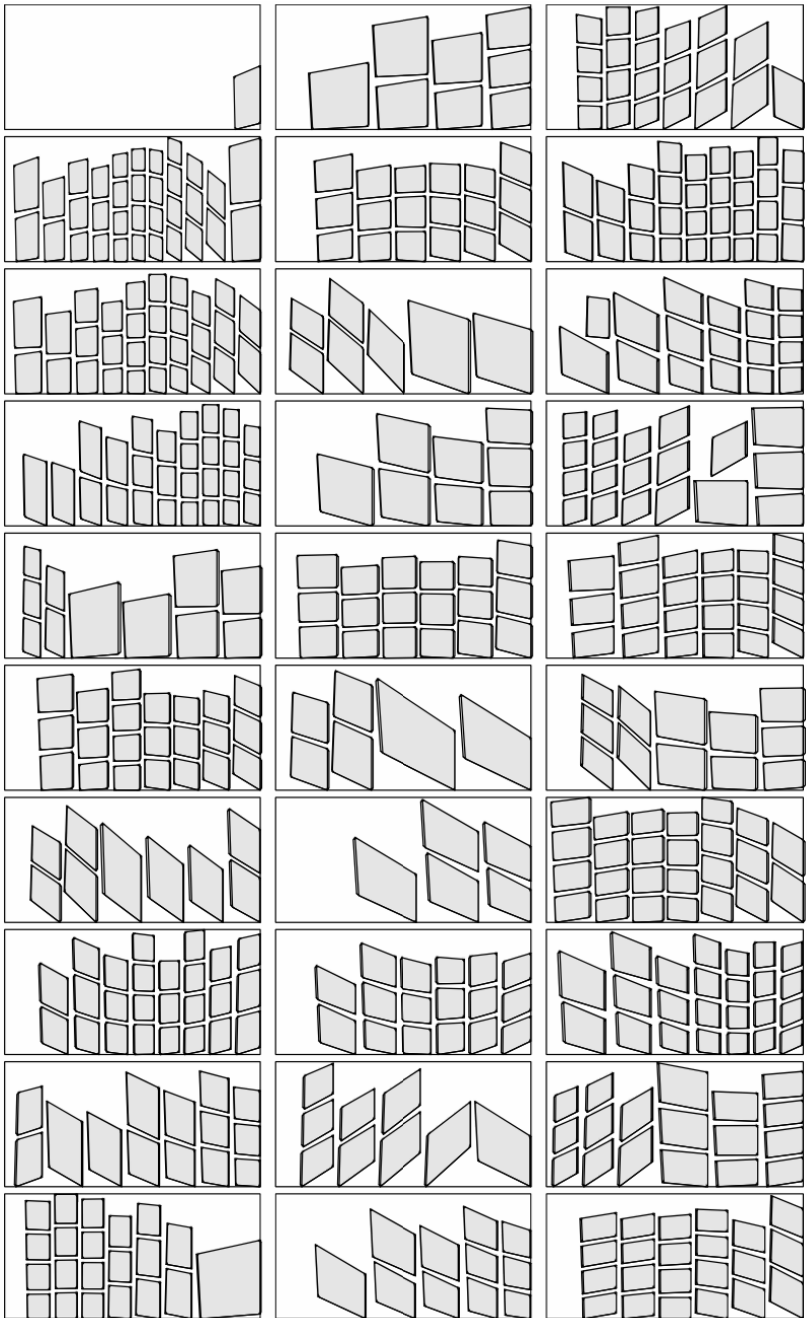


FIGURE 8.4.11. Production plans of prototype #4: 30 raw material panels and 512 constructional elements

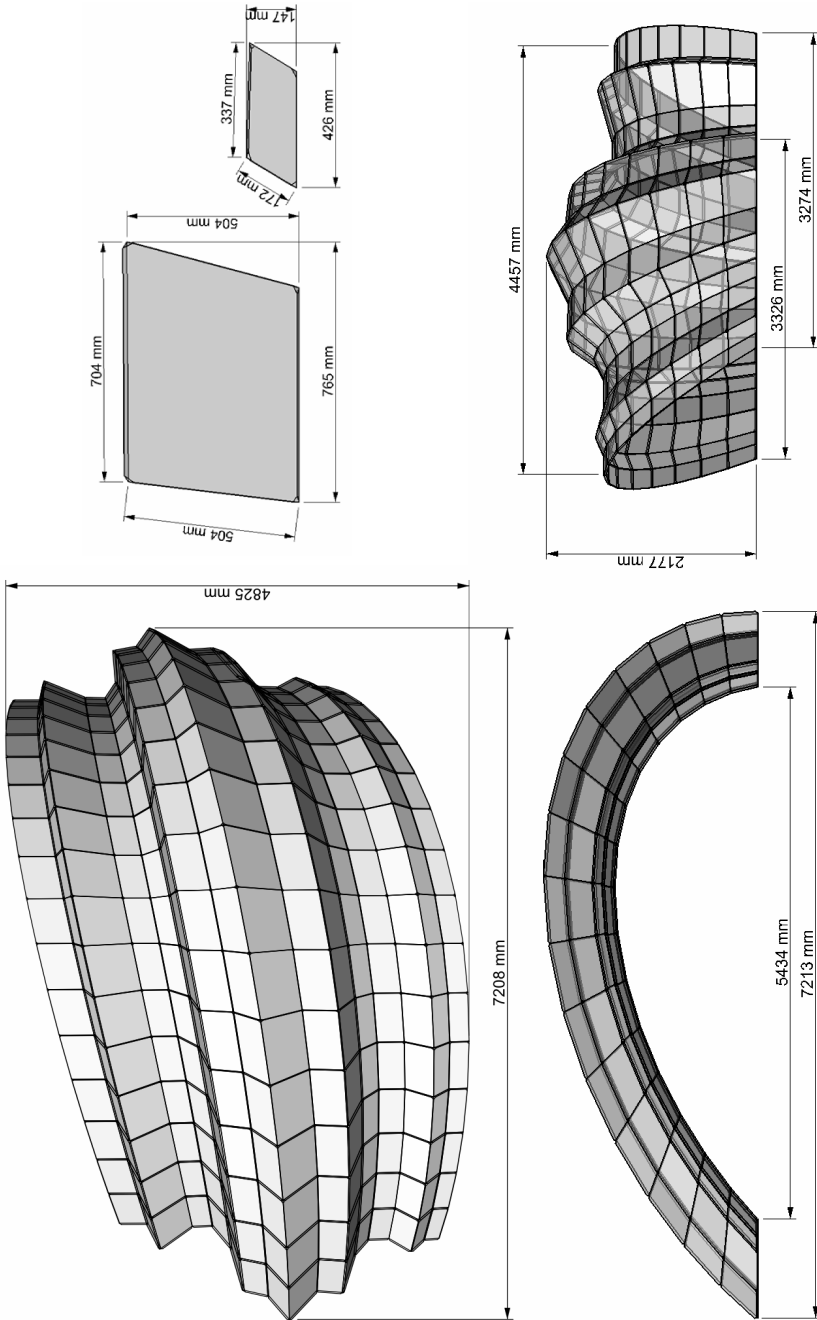


FIGURE 8.4.12. Design of prototype #5

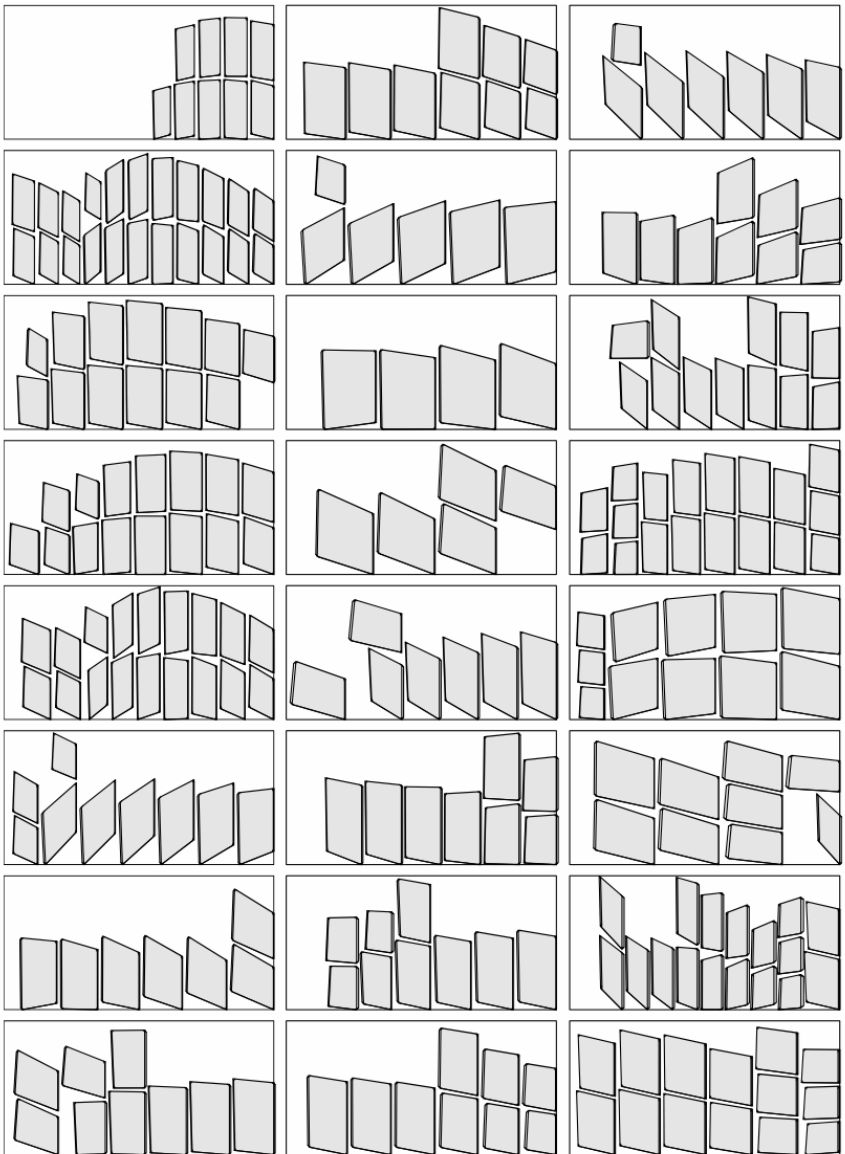


FIGURE 8.4.13. Production plans of prototype #5: 24 raw material panels and 256 constructional elements

8.5. Discussion

The shown examples explore a small part of the possibilities offered by iterative geometric design. The wide range of applications illustrate possible domains of applications of iterative geometric design. The realized prototypes have shown the advantage of a close relation between the geometrical model and the production process to design and to fabricate geometrically complex free-form architecture. Once a coherent work chain has been established, changes can be operated smoothly at any level of the design process. This has been shown particularly by the case study of the timber panel structures. After the validation of the design process, we were able to produce several differently shaped and sized prototypes. The examples show that changes on the form finding level do not imply all the detailing and workshop plans to be reworked. Analogous to this, other parameters like for instance material thickness or detailing can be changed independently. This provides greater flexibility for the design and the production of geometrically complex free-form architecture.

Part 4

Conclusion and Outlook

CHAPTER 9

Conclusion

9.1. Discussion

Within this thesis, several notions related to digital design and production of architecture have been presented. Undertaken investigations range from the research of new geometric architectural design methods to optimizations of integrated manufacturing techniques. The spectrum of the research has intentionally been widely spread in order to provide a global approach to the problem of design and production of architectural free-form objects. It is highly important to situate the single links of the digital chain within the global context of the design process engaged.

The applications presented in chapter 8 show how beneficial and time-saving the use of a complete design chain can be, allowing to intervene independently at any of the design or production steps and permitting changes within the design process to be integrated smoothly. The initial appraisal - claiming the lack of feasibility of free-form structures¹ - shows the importance of the optimization of design and production processes involved within the realization of architectural free-form objects.

At the level of geometric design, the findings and contributions discussed below are not specific to architecture and can therefore be applied to other domains. Going down to the level of production, the solutions become more and more specific as they are developed in function of particular architectural designs². They establish a close relation between the initial geometrical figure and its built counterpart. The specificity of the methods depending on a given project provides functional, efficient, light and rapid tools. However, the methods worked out on the level of production are less specific than the ones on the level of geometric design.

¹mainly due to tremendous production and planning effort

²which are depending on the type of application, contextual setting, material, manufacturing device, etc.

9.2. Contributions

Within the following the contributions to existing work and methods are discussed with regards to their respective domains.

9.2.1. Iterative Geometric Design.

- (1) *Iterative geometry* unifies within one formalism the methods of fractal modeling (IFS), subdivision geometric design and L-systems. The general geometric design formalism provides iterative construction of a very large variety of geometric figures unifying the up to now separate shape grammars of smooth and rough figures. Hence, architectural geometric design is extended by the world of fractal and folded shapes.
- (2) *Transformation driven geometric design* exposed in section 4.1 provides an intuitive manner to manipulate the transformation matrices. They specify the local aspect of a geometric figure characterizing the level and the nature of its respectively roughness and smoothness. Introducing subdivision point modeling, the shape defining parameters can be ascertained graphically. Additionally, exact control is provided by declarative input methods. Besides subdivision point modeling, classical design methods such as control point editing are offered by transformation driven iterative geometric design.
- (3) *Constraints* are handled topologically and geometrically. They are of high importance to architectural geometric design allowing to formulate criteria needed for the later physical realization of the generated geometrical figures. Topological constraints provide control over connectivity and continuity of the components of the figure being designed. Geometrical constraints have been integrated in the surface method (cf. section 4.3) verifying that all components of a free-form surface remain planar in any case.
- (4) *Discrete architectural geometry* provides the advantage of free-form figures being expressed by a finite number of elements. The number of elements is appointed by the level of iteration used for the construction of a discrete figure. Hereby, control over the required refinement and level of detail of a figure designed is provided. Further, the discrete expression delivers a modular raster being able to be used as a construction grid determining a finite set of components potentially translatable into constructional elements.

9.2.2. Generation of Constructional Elements of free-form architectural objects. The generation of the data describing the constructional components of iterative geometric objects raised methods for data structuring. Those allow the set of geometric data to be conditioned for processing with regards to integrated manufacturing. Findings in this domain are resumed within the following areas:

- (1) *Addressing systems* give access to every individual member of the global object. Unique IDs are associated to each element in an ordered system. The addressing system provided by the IFS geometric design method establishes an inherent relation between the geometric figure and the data structure of the component's IDs. Beyond the advantages and the limits of IFS-addressing (cf. section 6.2), lexicographically ordered addressing systems can be used in a complementary way.
- (2) *Geometrical Description of the Constructional Components* is computed in function of the geometry data of an iterative figure. Border conditions of architectural free-form objects differentiate the elements within the global structure of the figure being designed. Parameters depending on the application and the material used for the physical realization of the object allowed to refine the method for translating the geometric components into a complete set of constructional elements.
- (3) *Automated Detailing* completes the global description of the constructional components. Different generic details may be applied in function of the respective requirements. The addressed data structure allows differentiating border details from inner connections, where again different fastening techniques may be used for respectively lateral and longitudinal joining.

Contextual conditions integrate into the translation method for the conversion of the geometric elements into constructional components. On the one hand, the data structure used is in direct relation to the general geometric design method. On the other hand, the methods deployed for the translation of the geometry data into constructional components have been tightly specialized with regards to their respective applications.

9.2.3. Integrated Manufacturing. Having defined the constructional elements, the parts are finally processed for integrated manufacturing. The following findings made it possible to directly process the data set describing the constructional elements into a set of machine instructions for integrated manufacturing.

- (1) *The Mapping method* allows the components being translated from the model space³ to the machine space⁴ with regards to integrated manufacturing. The data structure describing the constructional components is enriched by local object coordinate systems proper to each component. The definition of individual object coordinate systems allows flawless mapping of the constructional elements from the model to the machine space offering control over the final orientation of the parts, with regards to the anisotropy of the used material.
- (2) *Packing methods* studied enhance the waste/raw material ratio. Although the developed method does not reach an optimal exploitation of the raw material, it actually achieves a fair trade off between packing time and waste material. It preserves the orientation of the components according to e.g. grain direction. Furthermore, the proposed method integrates parameters from the level of machining such as machining strategies or tool geometry. Hereby, a packing solution for 5-axis multi-part manufacturing is provided which does not exist at present.
- (3) *Automated Creation of G-code* completes the digital production chain with regards to integrated manufacturing. The process used for the generation of the set of machine instructions has been optimized in accordance to the nature of the components being produced. Machining strategies depending on the part's geometry and the connection details allow the generation of adequate tool paths, that permit minimizing manufacturing time. On the one hand, this optimization may limit the generality of the method. On the other hand, it allows automating the generation of tool paths and the post processing of the G-code files needed for the milling of the constructional elements. This can actually not be accomplished in an automated way by using existing CAM-solutions.

³Design Environment

⁴Production Environment

9.3. Contributions to the State of the Art

Contributions to the field of geometric architectural design are innovative and provide unseen form finding possibilities adding fractal modeling and a new constrained surface method to the design tools of common CAD-software. Iterative geometric design is general enough to be applied not exclusively to the field of architecture. Fractal modeling as defined by Tosan et al. [ZT96, TZTV97] has been enriched by subdivision point editing. The use of the projective geometry extends the method to rational figures, providing further form finding possibilities.

On the level of integrated production, more specialized methods have been developed in accordance to contextual and material conditions. The specificity of the developed methods - which is in direct relation to particular architectural designs - allows an efficient optimization of the digital chain for the design and the production of iteratively designed free-form objects. The provided links between the individual work steps fit closely allowing designers to intervene and operated changes at any level of the design cycle.

The wide spectrum of the interdisciplinary research permitted to integrate conditions from one field to the other. Cross relations of apparently intrinsic properties of the different domains were used to formulate conditions leading to new solutions.

First, material properties and machining techniques led to the definition of constraints that appeared to be fundamental for the development of a packing method for 5-axis multi-part manufacturing.

Second, addressing systems were designed in accordance with the geometric design methods. Additionally, they integrate conditions of the architectural context and properties defined by fastening and montage techniques.

Further, the property of iterative geometric design delivering discrete figures was used for automating the generation of constructional elements.

Hereby, properties of a mathematical geometric construction method were used to elaborate solutions to the problem of physical construction.

CHAPTER 10

Outlook

10.1. Applications

The applications of iterative geometric design discussed in chapter 8 show a wide range of applications. The presented examples have been empirically developed and represent only a fraction of possible future applications of iterative geometric design for architectural use.

Although the shown examples remain up to now relatively small in scale, they made it possible to verify the validity of the proposed design method. Hence, the employed manufacturing techniques are applicable for real scale constructional elements. The validity of the methods developed within the scope of this thesis is further endorsed by the fact that partial results in the field of integrated manufacturing were able to be applied on third party projects. A short description of projects using our methods can be found in the appendix of this document. The out-sourced findings have been applied within the following scopes:

- (1) *Integrated manufacturing techniques* developed within the scope of this work have been applied for the realization of different timber panel structures. The realizations range from the model making of students projects, presented in 11.3.1, to the machining of the parts for a wooden origami structure, shown in 11.3.2. The use of our methods has allowed efficient production of two complex architectural designs composed out of hundreds of differently shaped components. These two projects have drawn profit of the developed methods for mapping, tool-path generation, machining strategies and G-code generation (post-processing).
- (2) Methods for *automated generation of constructional elements* have been applied on two different projects: Roof design of a high-school's gymnasium, shown in the appendix under 11.2 and the wooden origami sculpture, presented in 11.3.2. The projects employed the developed data conditioning methods for addressing, thickening and generic detailing. In the example

of the roof design (cf. 11.2), the developed method was able to handle ten thousand elements. The constructional components were described by a complete set of data containing: The workshop plans of every single element, addresses and number listings, quantitative values indicating the amount of raw material needed and the number of cutting and drilling operations required for the production of the elements.

- (3) *Post-Processors*: While two projects [11.3.2 and 11.3.1] have used the post-processor developed for a 5-axis milling machine, the research carried out within chapter 7 led to the development of two 3-axis CNC-machine post-processors, presented in 11.3.3. The capability to adapt the post-processor to different CNC-devices is important as it shows the portability of the deployed G-code generation method with regards to the large variety of CNC-machines used in the industry.

Some of the results of this work were able to be applied within a broader and more general context as mentioned above. This opens perspectives for future development and future applications.

10.2. Future Work

10.2.1. Research on Iterative Geometric Design. Concerning constrained geometric design for architectural use, a new deterministic surface method (cf. 4.3) has been developed, which verifies all parts being planar. This constraint is of high importance for the construction of free-form architecture that are built out of flat sheet material.

While the proposed method is verifying its parts being planar, it does not guarantee the generated mesh figure being conical. The constraint of conical vertices is important for thickening of the geometric figure in order to compute the constructional components' geometry. Pottmann et al. [PLW⁺07, PW08] studied the problem and provide techniques for optimizing quad-meshes in order to meet the requirements. In addition to future research on optimization methods, it would be highly desirable to develop deterministic design methods providing directly planar and conical quad-meshes.

10.2.2. Development in Integrated Architectural Design. New tools for design and production of geometrically complex architecture have been developed within the scope of this work with regards to specific architectural designs. It is insofar important to understand that precise and efficient solutions may only be obtained by specifically designed tools. At present, a series of specific software tools implements the methods worked out (cf. 11.5).

For particular applications as the ones discussed above (cf. section 10.1), the developed tools need to be adapted depending on the project on which they are applied to. Therefore, adaption and expansion of the presented solutions with regards to future applications will continue to be an important field of development. Hence, the acquired knowledge and the gained experience will facilitate future developments.

10.2.3. Future Applications. The studied geometric design techniques are not restricted to strictly formal and ornamental applications, even though a completely new shape grammar has become accessible by iterative geometric design. The possibility of adding roughness and folds to architectural free-form designs may not only be used for aesthetic reasons but may actually be useful for structural applications too. A comparison of the application of the B-spline Shell (cf. 8.3) to the timber panel structure (cf. 8.4) shows how the introduction of folds provides higher bending rigidity to the overall structure. Preliminary structural analysis has been undertaken by Natterer (cf. appendix 11.4). Therefore, we consider applications of IFS-surfaces for load bearing structures to be an important future field of research.

In addition to this, iterative geometric design may be applied for the design of free-form facades, interior design, climbing walls or suspended ceilings.

Epilogue

Techniques and design methods of architects change over time. With the development of computer technologies and the arrival of computer aided design in the field of architecture new design possibilities emerged, providing access to advanced geometric design methods. It is important to state that today's architects often empirically employ the new tools without having a proper understanding about the implied mathematical concepts. Thus, the potentially powerful new technologies remain too often unmastered, which is hindering the deployment of their full capacity due to lack of control.

The present work take a critical look behind the scenes of these new technologies, making methods of computer aided design and production more understandable. The acquired knowledge enabled us to better form a more discerning opinion upon the underlying principles. Hence, computer aided tools may actually be used more accurately with regards to applications in the field of architecture. Further, most of the offered solutions are not directly applicable to the architectural realm. More specialized tools need to be developed in function of a specific architectural design. To succeed, the "digital" architect has to design his proper adequate and specific tools.

The interdisciplinarity of the research topic and the research team was key to accomplish that goal. It allowed to learn and exchange knowledge of geometrical design, software development, data structuring and production techniques. This knowledge is more and more important in the context of a changing profession. Applications as the projects shown in the appendix witness of how the acquired knowledge may directly contribute to the development of computer aided architectural designs, leading to a new design paradigm. This influences the way how architectural problems are approached, developed and solved. Therefore, we estimate that the developed topics support the foundation of a new generation of architects. Within this perspective, the different topics studied during the elaboration of this work have sustainably influenced to the author's work method and will surely be applied in the future.

Bibliography

- [Bar88] Michael Barnsley. *Fractals Everywhere*. Academic Press, 1988.
- [BL94] M. Batty and P. Longley. *Fractal cities: a geometry of form and function*. Academic Press, 1994.
- [Bov03] C. Bovill. *Fractal Geometry in Architecture and Design*. School of Architecture, University of Maryland, Boston, 2003.
- [BS96] Carole Blanc and Christophe Schlick. Accurate parametrization of conics by nurbs. *IEEE Comput. Graph. Appl.*, 16(6):64–71, 1996.
- [BW06a] Hani Buri and Yves Weinand. BSP Visionen - Faltwerkstrukturen aus BSP-Elementen. In *Grazer Holzbau-Fachtage*, 2006.
- [BW06b] Hani Buri and Yves Weinand. Origami: Faltstrukturen aus Holzwerkstoffen. *Bulletin Holzforschung Schweiz*, (2):8–12, 2006.
- [BW07] Hani Buri and Yves Weinand. Uebersicht Massivholzplatten. In *39. Fortbildungskurs Schweizerische Arbeitsgemeinschaft für Holzforschung*, pages 63–84, Dübendorf, 2007. SAH Schweizerische Arbeitsgemeinschaft für Holzforschung.
- [BW08a] Hani Buri and Yves Weinand. Die provisorische Kapelle von St. Ioup. *Bulletin Schweizerische Arbeitsgemeinschaft für Holzforschung SAH*, (2):16–20, 2008.
- [BW08b] Hani Buri and Yves Weinand. ORIGAMI - Folded Plate Structures, Architecture. In *10th World Conference on Timber Engineering*, 2008.
- [BW09] Hani Buri and Yves Weinand. Gefaltet. *TEC21*, (8):18–22, 2009.
- [Béz62] P. E. Bézier. *Numerical Control: Mathematics and Applications*. Wiley, 1962.
- [Can84] Georg Cantor. *De la puissance des ensembles parfaits de points*. Acta Mathematica 4, 1884.
- [CS93] J. H. Conway and N. J. A. Sloane. Sphere packings, lattices, and groups. *Proc. Phys. Math.*, 2nd ed, 1993.
- [dC59] Paul de Casteljaou. *Courbes à pôles*. INPI, 1959.
- [dC63] Paul de Casteljaou. *Surfaces à pôles*. INPI, 1963.
- [Dij99] Dirk Luberth Dijkman. *HP-GL, Graphics Language*. <http://www.luberth.com>, 1999.
- [Dür38] Albrecht Dürer. *De Symmetria... and Underweysung der Messung*. The Warnock Library, 1538.
- [Dür77] Albrecht Dürer. *The Painter's Manual*, translated by Walter L. Strauss. Abaris Books Inc., 1977.
- [Fal90] K. Falconer. *Fractal Geometry, Mathematical Foundations and Applications*. Wiley, 1990.

- [Fra94] P. Frankhauser. *La fractalité des structures urbaines*. Anthropos, 1994.
- [Fra97] P. Frankhauser. *L'approche fractale: Un nouvel outil de réflexion dans l'analyse spatiale des agglomérations urbaines*. Population, p.1005-1040 no.4, 1997.
- [Gen92] Christian Gentil. *Les fractales en synthèse d images: le modele IFS. Thèse de doctorat*. Université LYON I, 1992.
- [GSC⁺04] James Glymph, Dennis Shelden, Cristiano Ceccato, Judith Mussel, and Hans Schober. A parametric strategy for free-form glass structures using quadrilateral planar facets. *Automation in Construction*, 13(2):187 – 202, 2004. Conference of the Association for Computer Aided Design in Architecture.
- [GTSW08] Gilles Gouaty, Eric Tosan, Ivo Stotz, and Yves Weinand. Un modèle itératif de surface pour la construction en bois. In *Groupe de travail en modélisation géométrique GTMG*, 2008.
- [GTSW09] Gilles Gouaty, Eric Tosan, Ivo Stotz, and Yves Weinand. *Un modèle itératif de surfaces pour la construction en bois*, volume 3. 2009.
- [HH89] K. Humpert and H.Bohm. *Natürliche Prozesse: Haus und Stadt*. SFB 230 Teil 2, Universität Stuttgart und Universität Tübingen, 1989.
- [Hil91] David Hilbert. *Über die stetige Abbildung einer Linie auf ein Flächenstück*, volume 38. Mathematische Annalen, 1891.
- [Hut81] Jhon E. Hutchinson. *Fractals and self-similarity*. Indiana Univ. Math. J, 30, 1981.
- [HW07] Marcel Haasis and Yves Weinand. Origamifaltwerke, Neue Anwendung für Brettsperrholzplatten. In *39. SAH Fortbildungskurs*, pages 175–181, Dübendorf, 2007. SAH Schweizerische Arbeitsgemeinschaft für Holzforschung.
- [HW08a] Marcel Haasis and Yves Weinand. ORIGAMI - Folded Plate Structures, Engineering. In *10th World Conference on Timber Engineering*, 2008.
- [HW08b] Marcel Haasis and Yves Weinand. Versuche an Verbindungen für Faltwerke aus Brettsperrholzplatten. In *Statusseminar Schweizerische Arbeitsgemeinschaft für Holzforschung*, 2008.
- [HW09] Markus Hudert and Yves Weinand. Tragendes Holzgewebe. *Holzforschung Schweiz Bulletin SAH*, (1):7–9, 2009.
- [Koc04] Helge Von Koch. *Une courbe continue sans tangente, obtenue par une construction géométrique élémentaire*, volume 1. Arkiv för Matematik 1, 1904.
- [LDFS04] A. Y. T. Leung, H. Dai, S. L. Fok, and R. K. L Su. *The fractal finite element method for unbound problems*. International Journal For numerical methods in Engineering, Wiley, 2004.
- [Lév93] Paul Lévy. *Plane or Space Curves and Surfaces Consisting of Parts Similar to the Whole*. Addison-Wesley Publishing, 1993.
- [Man82] B. Mandelbrot. *The fractal geometry of nature*. Freeman, 1982.
- [Min96] H. Minkowski. *Geometrie der Zahlen*. Teubner, 1896.
- [Mon47] Gaspard Monge. *Géométrie descriptive*. 1847.
- [Mon50] Gaspard Monge. *Application de l'analyse à la géométrie*. 1850.
- [MP87] Charles A. Micchelli and Hartmut Prautzsch. Computing surfaces invariant under subdivision. *Computer Aided Geometric Design*, 4(4):321–328, 1987.

- [MP02] E.S. Mistakeidis and O. K. Panagouli. *Strength evaluation of retrofit shear wall elements with interfaces of fractal geometry*. Engineering Structures 24, 2002.
- [NW] Johannes Natterer and Yves Weinand. Modeling of multi-layer beam with interlayer slips. In *10th World Conference on Timber Engineering*.
- [NW07] Johannes Natterer and Yves Weinand. Sechsteiliges, nachgiebiges Verbundelement - Tragverhalten. *Bulletin Holzforschung Schweiz*, (1):15–17, 2007.
- [PBCW07] Helmut Pottmann, Sigrid Brell-Cokcan, and Johannes Wallner. Discrete surfaces for architectural design. In Patrick Chenin, Tom Lyche, and Larry L. Schumaker, editors, *Curves and Surface Design: Avignon 2006*, pages 213–234. Nashboro Press, 2007.
- [PLW⁺07] Helmut Pottmann, Yang Liu, Johannes Wallner, Alexander Bobenko, and Wenping Wang. Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graphics*, 26(3), 2007. Proc. SIGGRAPH.
- [PM87] Hartmut Prautzsch and Charles A. Micchelli. Computing curves invariant under halving. *Computer Aided Geometric Design*, 4(1-2):133–140, 1987.
- [PolBC] Marcus Vitruvius Pollio. *De architectura*. <http://la.wikisource.org/wiki/>, 40 BC.
- [Pru89] P. Prusinkiewicz. *Lindenmayer systems, fractals and plants*. Springer, New York, 1989.
- [PW06] Claudio Pirazzi and Yves Weinand. Beschreibung geodätischer Netze auf Freiformflächen: Entwurf und Konstruktion eines Prototyps zur Evaluierung eines Berechnungsprogramms. In *Kolloquium Holzbau*, 2006.
- [PW08] Helmut Pottmann and Johannes Wallner. The focal geometry of circular and conical meshes. *Adv. Comp. Math*, 29:249–268, 2008.
- [RR03] S. L. Rodrigez and D. J. Lopez Rodrigez. *Geometria fractal en la Alhambra*. p.37-61, Publicaciones del patronato de la Alhambra y Generalife, 2003.
- [Sal02] Nicoletta Sala. *The presence of the Self-similarity in Architecture: some examples*. Emergent Nature - Patterns, Growth and Scaling in the Sciences, p.273-282, World Scientific, 2002.
- [SGW08] Ivo Stotz, Gilles Gouaty, and Yves Weinand. Iterative Flächengestaltung. *Bulletin Holzforschung Schweiz*, (1):29–34, 2008.
- [SS02] J. Schlaich and H. Schober. *Filigrane Kuppeln: Beispiele, Tendenzen und Entwicklungen*. vol.128, no.12, p.21-27, Tec21, 2002.
- [SWG08] Ivo Stotz, Yves Weinand, and Gilles Gouaty. Iterative surface design for constructions based on timber panels. In *Statusseminar Schweizerische Arbeitsgemeinschaft für Holzforschung SAH*. EMPA, 2008.
- [Tak03] Teiji Takagi. A simple example of a continuous function without derivative. *Proc. Phys. Math.*, 1:176–177, 1903.
- [TBSG⁺06] Eric Tosan, Iver Bailly-Salins, Gilles Gouaty, Ivo Stotz, Peter Buser, and Yves Weinand. Une modélisation géométrique itérative basée sur les automates. In *Groupe de travail en modélisation géométrique GTMG*, 2006.
- [TGB02] Eric Tosan, Eric Guerin, and Atila Baskurt. *Design and reconstruction of fractal surfaces*. In IEEE Computer Society, editor, 6th International Conference on Information Visualisation IV, 2002.

- [Tho96] Joëlle Thollot. *Extension du modèle IFS pour une géométrie fractale constructive. Thèse de doctorat*. Université LYON I, 1996.
- [Tos96] Eric Tosan. *Une Algèbre de Formes fractales utilisant les IFS. Thèse de doctorat*. Université LYON I, 1996.
- [Tos99] Eric Tosan. *Wire frame Fractal Topology and IFS Morphisms*. 4th Conference Fractals in Engineering, 1999.
- [Tri89] Kirti Trivedi. *Hindu temples: models of a fractal universe*. The Visual Computer, p.243-258 no.5, Spinger-Verlag, 1989.
- [TSGW07] Eric Tosan, Ivo Stotz, Gilles Gouaty, and Yves Weinand. Modélisation itérative de courbes et surfaces: aspect multi-résolution. In *Groupe de travail en modélisation géométriques GTMG*, 2007.
- [TZTV97] Joëlle Thollot, Chems Eddine Zair, Eric Tosan, and Denis Vandorpe. *Modeling fractal shapes using generalizations of IFS techniques*. In Jacques Lévy Véhel, Evelyne Lutton, and Claude Tricot, *Fractals in Engineering*, 1997.
- [VG95] N. Vasilyev and V. Gutenmacher. Dragon curves. *Quantum*, 6:5–10, 1995.
- [vO90] Cornelius W. A. M. van Overveld. Family of recursively defined curves, related to the cubic bézier curve. *Computer-Aided Design*, 22(9):591–597, 1990.
- [Wei08a] Yves Weinand. Innovative Timber Construction. In *10th World Conference on Timber Engineering*, 2008.
- [Wei08b] Yves Weinand. New Applications in Timber Construction. In *World Conference in Timber Engineering*, 2008.
- [WP06] Yves Weinand and Claudio Pirazzi. Geodesic Lines on Free-Form Surfaces - Optimized Grids for Timber Rib Shells. In *World Conference in Timber Engineering WCTE*, 2006.
- [Zai98] Chems Eddine Zair. *Formes fractales à pôles basées sur une généralisation des IFS. Thèse de doctorat*. Université LYON I, 1998.
- [ZT96] Chems Eddine Zair and Eric Tosan. *Fractal modeling using free form techniques*. Computer Graphics Forum, 15. EUROGRAPHICS-96 Conference, 1996.

List of Figures

2.2.1	Example of a 5-Axis milling machine	22
3.1.1	Pythagoras tree	29
3.2.1	Cantor set	30
3.2.2	Von Koch curve	31
3.3.1	Iterative construction of a Bézier-curve	31
3.3.2	Barnsley's illustration for IFS	32
3.3.3	Graphical representations of the Attractor as a union of its components	33
3.3.4	Smooth and fractal figure with control points designed by projected IFS according to [ZT96]	33
4.1.1	First construction steps of a Bézier curve	35
4.1.2	Construction sequence: Tree representation	36
4.1.3	Representation of control and subdivision points	38
4.1.4	Relation between the control points P, the subdivision points S and the transformations matrices T	38
4.1.5	Acting on the transformation matrices by manipulating the subdivision points (shown in blue)	39
4.1.6	IFS curve design: adjustment of roughness / smoothness	40
4.3.1	Centered vector sum surface	42
4.3.2	Deformation of projected vector sum meshes by editing the weight of certain points	43
4.3.3	Point weight editing of smooth and rough projected IFS vector sum surfaces	44
4.3.4	Example of a Vector sum Surface	45
5.1.1	Cantor Set	48

5.1.2	Un-constrained Cantor Set	49
5.1.3	Two-dimensional Cantor Set	50
5.1.4	Cantor set variations	51
5.1.5	Coinciding subdivision points: Continuous IFS-curves a) and b), plane filling IFS-object c)	51
5.1.6	Self-intersecting fractal objects	52
5.2.1	Continuous IFS-curve	53
5.2.2	Rational quadratic Bézier curves: Iterative construction steps $n = f1;2;3;4g$; influence of point weight editing	54
5.2.3	Representation of conic curves using multiple patches of discrete IFS-curves. The numbers shown in the figure correspond to the w -values of the respective control points.	55
5.2.4	Subdivision point editing	57
5.2.5	Subdivision point modeling providing variations of an IFS-curve	59
5.2.6	Cubic Bézier curve	60
5.2.7	a) Iteratively designed cubic Bézier curve; b) Cubic Takagi curve	60
5.2.8	Cubic B-Spline	61
5.2.9	Subdivision point editing of a cubic B-Spline curve shown in the middle column	62
5.3.1	(a) $p1 = s1; s3 = s4; s6 = p3$ (b) $p1 = s1; s3 = s6; s4 = p3$ (c) $p1 = s1; p2 = s3 = s4; s6 = p3$ (d) $p1 = s1; p2 = s3 = s6; s4 = p3$	63
5.3.2	The orientation of the line segments influences the resulting figure	65
5.4.1	Examples of iteratively designed surfaces	66
5.4.2	3x3 tensor product surface: Bézier-patch on the left, fractal-patch on the right	67
5.4.3	Multi-patch surfaces: a) One, b) two and c) four patches	69
5.4.4	Fractal cylinder designed by multi-patch vector sum surfaces	70
5.4.5	Vector sum surface by two curves of different levels of iteration	71
6.2.1	Construction tree and relating addresses	77
6.2.2	Ordered one dimensional list	78
6.2.3	Ternary addressing system induced by the three used transformations	79
6.2.4	The orientation of the curve affects the addressing order.	79

6.2.5	IFS-addresses of the Sierpinski triangle	80
6.2.6	Addressing systems	82
6.2.7	Valances of quad meshes with regards to the number of neighbors	83
6.2.8	Quad mesh topology and face types (B = Bottom, L = Left, R = Right, T = Top, C = Center)	83
6.2.9	Ordering mesh faces of a quad mesh figure: a) arbitrary b) linear list c) two-dimensional array	84
6.2.10	Face-Edge-Vertex order of polygonal mesh figures	85
6.2.11	Corresponding vertices of adjacent quad mesh faces.	85
6.3.1	Description of a quad panel	87
6.3.2	Face offset: generation of the vertex v'_4	88
6.3.3	Imaginary neighbors of border and corners mesh faces	89
6.3.4	Applications of free form quad meshes: a) bearing structure b) suspended ceiling	90
6.3.5	Normal vectors of the border mesh face, the limit surface and the imaginary neighbor	90
6.3.6	Thickening of a folded plate structure	91
6.3.7	Thickening of a planar quad mesh surface	92
6.3.8	Angle criterion according to [PW08]	93
6.3.9	Offset problem on non-conical vertices	93
6.3.10	Resulting constructional elements of conical and non-conical meshes respectively	94
6.4.1	Example of a set of linearly arrayed fasteners along the edges of a quadrilateral timber panels	96
6.4.2	Parametrical representation of the detail of one side of a quad panel according to figure 6.4.1	97
6.4.3	Mechanical fastener for joining panel at various angles	99
6.4.4	Different fasteners for the assembly of timber panels: Hinge joints {d}, hard wood dowels {e}, mechanical fasteners {a,b,c,f}	100
6.4.5	Panel with pre-drilled holes for hard wood dowels	101
6.4.6	Montage problem with dowel type fasteners	102
6.4.7	Row montage of quad mesh structures	103
7.1.1	World and object coordinate system	106

7.1.2	Rotations mapping the object coordinate system collinearly to the world coordinate system	107
7.2.1	Pseudo 2D-description of two constructional elements.	109
7.2.2	Packing P2 with regards to P1	110
7.2.3	Packing one complete container	111
7.3.1	Maka MM 7s	112
7.3.2	Pocketing	114
7.3.3	Contour cut using a circular saw blade	114
7.3.4	Contour cut using a flat tipped finger tool	115
7.3.5	Tool paths for the manufacturing of one chamfered timber panel	116
7.3.6	Section view of tool path offset generation	116
7.3.7	Axonometry view of tool path offset generation	117
7.3.8	Rectangle containing the chamfer surface	118
7.3.9	{A;B}-angles from vector	120
7.4.1	CNC-machined MDF quad elements	121
8.1.1	Construction principle of the Hilbert curve	124
8.1.2	Hilbert curve parameters	124
8.1.3	Hilbert curve variations	125
8.1.4	Hilbert tilings 1	126
8.1.5	Hilbert tilings 2	127
8.1.6	Dürer's Pentagon	128
8.1.7	Parametrized pentaflake	128
8.1.8	Two prototypes of fractal shading panels	129
8.2.1	Iterative Bézier Curve	130
8.2.2	Shape study	131
8.2.3	Relation between the geometry data of the discrete Bézier curve (shown in red) and the workshop drawing of one constructional element	131
8.2.4	Workshop plans of one layer of the Bézier vault	132
8.2.5	Reduced Scale Model	132
8.2.6	Construction Detail of the Bézier Vault Structure (top and front view)	133
8.3.1	Four construction steps of a B-Spline-surface	134

8.3.2	Non planar quads present two ways of triangulation. Depending on the angle value between the triangular planes the case presenting the lower curvature is chosen. This allows lessening the initial stress of the bent timber board.	135
8.3.3	Quadrilateral faces are generally non planar. To unroll the shown quad {A;B;C;D}, point C is transformed by a rotation along the axis [B;D], in order to be part of the plane {A;B;D}.	136
8.3.4	Unrolled cutting pattern: The Length of the sides and two opposite angles are preserved.	136
8.3.5	Non planar quads present two ways of triangulation. The rendering of the shape is a more or less accurate representation of the initially modeled geometric object.	137
8.3.6	Tensor product surface of two cubic B-splines. [Left] Curvature analysis showing the greatest curvature near the corners of the shape. [Right] Geometric object after planarity optimization.	138
8.3.7	Each board, different in size and shape, receives a unique address. Automated generation of the CNC-files are prepared for integrated production and produced by a 3 axis CNC-machine.	139
8.3.8	The method and the geometry are tested and verified on a reduced scale prototype	140
8.4.1	Top and side views of the surface design	141
8.4.2	Parallel Offset Mesh Generation	142
8.4.3	Thickened ISF-surface	142
8.4.4	Integrated manufacturing of the constructional elements	143
8.4.5	Assembling of the partial prototype	145
8.4.6	Machined elements of the second prototype	145
8.4.7	Workshop plans of the third prototype	146
8.4.8	Second prototype - 8 by 8	147
8.4.9	Interior and exterior views of the third prototype, which is constituted of 256 elements	148
8.4.10	Design of prototype #4	150
8.4.11	Production plans of prototype #4: 30 raw material panels and 512 constructional elements	151
8.4.12	Design of prototype #5	152

8.4.13	Production plans of prototype #5: 24 raw material panels and 256 constructional elements	153
11.2.1	Design of the architects: double curved roof	180
11.2.2	Assembly principle of geodesic timber grid shells according to [NW07]	181
11.2.3	Grid lines: top and side view	181
11.2.4	Grid shell detail scheme	182
11.2.5	Multilayer model with labelled grid lines	183
11.2.6	Top: Geometrical description of one layer of one third of the roof. Bottom: Detail view	184
11.3.1	Generated tool-paths (shown in red) and the components (shown in grey)	188
11.3.2	Assembly of the student's model	189
11.3.3	Left: Initial surface; Right: Thickened components	190
11.3.4	Mapped and packed parts	191
11.3.5	Production of the eye-catcher's parts	191
11.3.6	Finished eye-catcher	192
11.3.7	Parameter input dialog	194
11.4.1	Comparison of the a smooth and rough IFS-surface under asymmetrical load	196

Nomenclature

2D	Twodimensional (generally in R2)
3D	Threedimensional (generally R3)
ALICE	atelier de la conception de l'espace
CAD	Computer Aided Design
CAGD	Computer Aided Geometric Design
CAM	Computer Aided Manufacturing
CNC	Computer Numerically Controlled
DEA	Diplôme d'études approfondies
ENAC	Faculté de l'Environnement, Naturel, Architectural et Construit
EPFL	Ecole Polytechnique Fédérale de Lausanne Swiss Federal Institute of Technology Lausanne
FEM	Finite Element Method
HDR	Habilitation à Diriger des Recherches
HP-GL	Hewlett Packard Graphic Language
IASS	International Association for Shell and Spatial Structures
ID	Identification
IFS	Iterated Function System
IGAT	Institut de Géométrie, Algèbre et Topologie
INSA	Institut National des Sciences Appliquées
IS	Intstitut de Structures
ISO	International Organization for Standardization
LIRIS	Laboratoire d'InfoRmatique en Image et Systèmes d'information
LMGC	Laboratoire de Mécanique et Génie Civil
LTH2	Laboratoire de Théorie et d'Histoire 2
MDF	Medium Density Fiberboard
NURBS	Non Uniform Rational B-Splines
RPM	Rotations Per Minute
SAR	Section d'ARchitecture (EPFL)

CHAPTER 11

Appendix

11.1. About this Appendix

The appendix features a series of projects, in which some of the methods studied within the scope of this PhD. thesis have been applied. The projects have consciously not been discussed in chapter 8 since no iterative geometric design has been employed to define the geometry of the architectural objects. However, with regards to integrated production of constructional elements of architectural free-form objects, following methods have been employed:

- (1) Integrated manufacturing techniques developed in chapter 7 have been applied for the fabrication of two timber plate structures (cf. 11.3.1 and 11.3.2).
- (2) Automated generation of constructional components developed in chapter 6 were used for two projects (cf. 11.2, 11.3.2).
- (3) Method for automated G-code generation led to the development of two HP-GL post-processors, which have been developed for the generation of machine instructions of two small scale 3-axis routers (cf. section 11.3.3).

These examples illustrate the capabilities of the developed methods to be applied more generally for the production of geometrically complex architectural designs. They show that findings of this thesis may be partially outsourced. Further, they provide an outlook on possible future development of our methods.

Additionally, load bearing applications of IFS timber panel surfaces are briefly addressed within the scope of section 11.4. A preliminary structural analysis evaluates some advantages of iterative geometric design for load bearing timber shell structures.

Finally, a note on different software implementations of the presented methods can be found in the end of this appendix, cf. 11.5.

11.2. Geodesic Rib Shell

11.2.1. Project Description. Within the following, a project of a triple gymnasium for a high school in Belgium will be presented. In March 2008, our laboratory IBOIS was asked to submit a constructive solution for the structure of the free-form roof designed by the architect. The part we worked on is shown in purple in figure 11.2.1. The dimensions of the freely spanning roof were 48 by 28 meters. We proposed to build it in form of a geodesic timber grid shell. Once the general layout of the structural grid has been designed, the question was about the creation of the workshop plans. The structural analysis, demanded an important cross section of the ribs, such that it was decided to work on a multi-layered model composed of eight layers of timber planks.

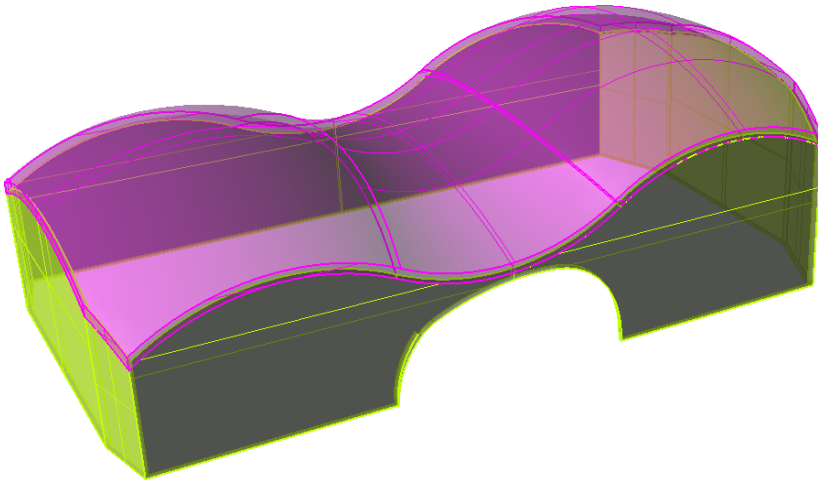


FIGURE 11.2.1. Design of the architects: double curved roof

11.2.2. Structural System. The principle of geodesic timber grid shells is to build free-form structures out of straight raw timber planks. The planks are bent onto the geometry of the surface describing a given architectural design. The layout of the bended timber planks form a grid. At the intersection of two grid lines, a bolt is placed in order to maintain the relative position of the planks. Depending on the required bearing capacity, the grid lines are composed of several layers of timber planks, which are nailed or screwed together in order to form a semi-rigid cross section (cf. figure 11.2.2). In terms of material continuity of multi-layer

timber grid shells, every second layer is constituted of continuous planks in one direction of the grid. Every other layer, the continuous grid lines are alternating the grid's direction. Discontinued grid lines are materialized in form of fill in planks.

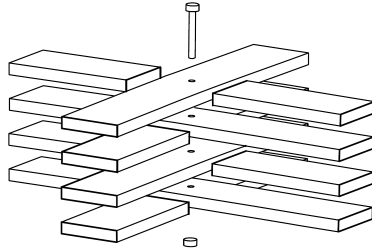


FIGURE 11.2.2. Assembly principle of geodesic timber grid shells according to [NW07]

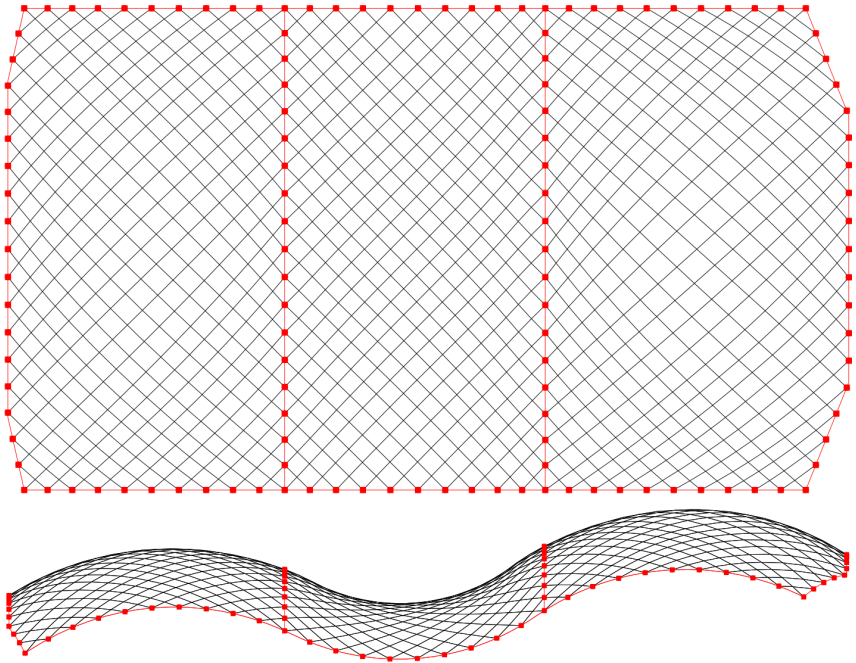


FIGURE 11.2.3. Grid lines: top and side view

Having defined the constructional system, the roof structure was divided into three parts. Thereon the subdivision of the surface could be

done, in order to obtain the rhythm and the position of the grid lines. Figure 11.2.3 shows the lay out of the grid lines. The red lines show the frame beams supporting the geodesic timber rib shell, defining the border conditions.

11.2.3. Parameters. In order to investigate the technical feasibility of the production of workshop data, one third of the roof structure was studied more in detail. With regards to integrated manufacturing of the constructional elements, the production conditions of a geodesic grid shell done by Pirazzy in 2005 have been considered. Detailed information about the realization and structural testing of this grid shell can be found in [PW06]. The construction principles match basically the ones presented within the aforementioned paragraphs.

Within the following, the relevant parameters for the design of the constructional elements are listed:

- Number of Layers : 8
- Width of the planks : 16 cm
- Height of the planks : 4 cm
- Joint dimensions : 0.2 cm

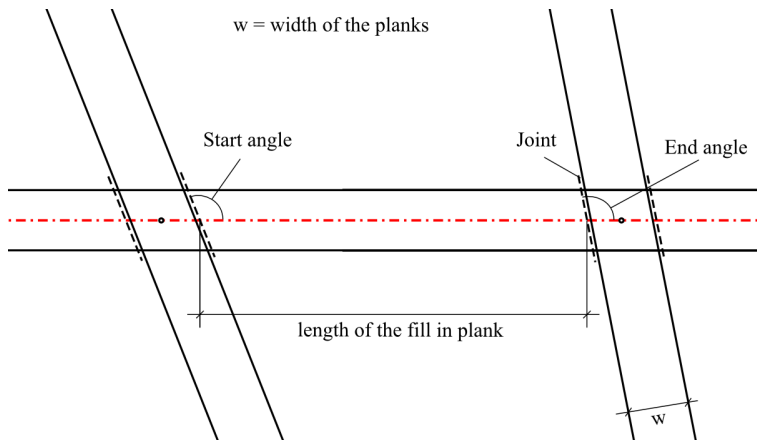


FIGURE 11.2.4. Grid shell detail scheme

The parameters are illustrated by figure 11.2.4. The Joint dimension of 2 mm is designed such that the fill planks are being produced

slightly smaller than the dimensions of the space in-between two continuous plank lines. This allows the easy insertion of the fill planks. In addition to this, the fill planks shall not provide direct contact to the continuous plank within the same layer.

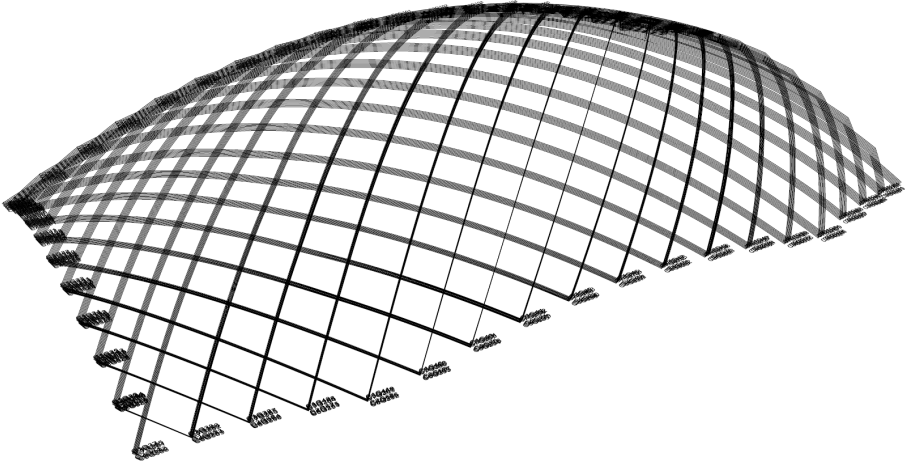


FIGURE 11.2.5. Multilayer model with labelled grid lines

By means of the number of layers and the planks thickness, the geometry of a multilayer model of the grid shell could be calculated. Figure 11.2.5 shows a perspective view of the multilayer model. Note: The different grid lines of the different layers have been labeled by unique IDs. These names will be used later for the element listings describing each single plank. Due to the amount of information existing at this stage of the design, graphical representation of the constructional elements is actually not able to render all the data integrally. Other ways of data representation had to be considered.

11.2.4. Resulting Constructional Elements. The data describing the constructional elements was designed for submission to the contractors in order to get a price offer. The data set was given as an example illustrating the capabilities to condition the production data in order to optimize the fabrication of the elements. Automated generation of the production data allowed to deliver precise quantity information on machining operations and used raw material, which we considered necessary for accurate pricing.

Finally, two representation methods have been specified, which describe the production data of the constructional elements:

- (1) A graphical representation of each plank, describing the geometry data of the finished elements.
- (2) A piece list, describing and quantifying the integral set of operations needed to machine the elements (piercing and cutting cycles).

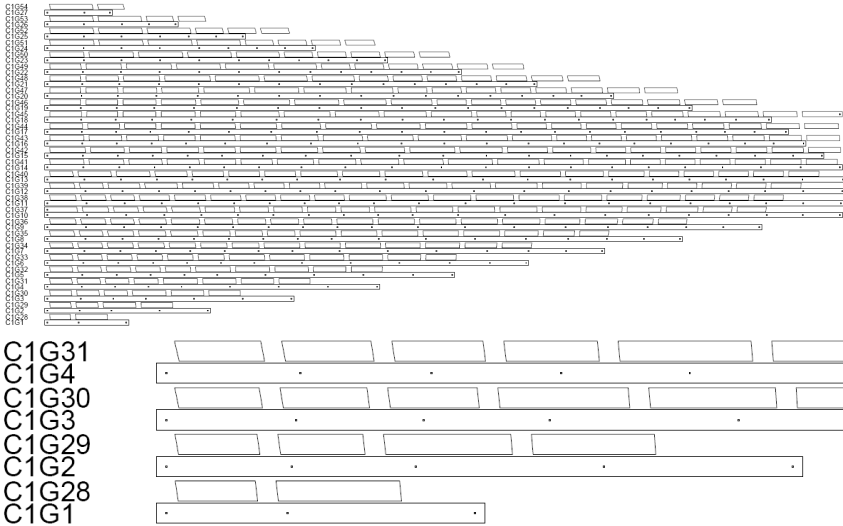


FIGURE 11.2.6. Top: Geometrical description of one layer of one third of the roof. Bottom: Detail view

Figure 11.2.6 shows the drawing describing the constructional elements of layer one. Each line of the drawing corresponds to one grid line of the rib shell. The name of the grid line is labeled at the left side of the drawing. Every second grid line is continuous and shows one plank over the whole span. The continuous planks present holes for the fixation of the bolts at the grid line's crossings. Every other line show a series of fill planks of one grid line. The fill planks do not present fixation holes but angle cuts at both of their ends. The values of the angle cut vary in function of the geometry of the grid line's crossings.

Within the following pages, an extract of the piece list is shown. The file starts with a summary of the employed parameters. Further each plank is described by the number of machining operations and the overall length of the element according to the scheme shown in figure 11.2.4. The

elements are sorted by layers, and by type {continuous, fill}. The continuous planks are further described by the number of holes to be drilled, while the fill planks are described by the chamfer cuts at both of its ends. At the end of the file, a summary of the used material and the number of pieces is plot, which may serve for the calculation of the production costs (allowing to differentiate material and machining costs).

```

Drawing Name : Geodesics1stThird
File creation date: 28.04.2008 09:20:46
-
V A R I A B L E S
Number of Layers      : 8
Width of the planks  : 16 cm
Height of the planks : 4 cm
Joint dimensions     : 0.2 cm
-
L A Y E R    1
Layer Name : layer1
-
Line C1G1  Continuous Plank
Length      : 245.772544296245
Number of holes : 3
-
Line C1G2  Continuous Plank
Length      : 498.751401188483
Number of holes : 5
-
Line C1G3  Continuous Plank
Length      : 758.405573682014
Number of holes : 7
-
Line C1G4  Continuous Plank
Length      : 1024.17148528496
Number of holes : 9
-
(...)
-
Line C1G30  Fill Plank :1
Length      : 66.9021892815474

```

```

Start Angle           : 101.517203999096
End Angle             : 99.5512888148112
-
Line C1G30  Fill Plank :2
Length               : 68.8834532022343
Start Angle         : 99.5512888148112
End Angle          : 97.7283146077738
-
Line C1G30  Fill Plank :3
Length               : 71.0382004399158
Start Angle         : 97.7283146077738
End Angle          : 96.0342480017642
-
Line C1G30  Fill Plank :4
Length               : 103.578372636248
Start Angle         : 96.0342480017643
End Angle          : 94.8557909417655
-
(...)
-
Line C8G432  Fill Plank :2
Length          : 62.5519549891013
Start Angle     : 83.8323675765027
End Angle      : 81.9254677456758
-
S U M M A R Y
Total Length   : 629872.277171699 cm
Number of elements : 3096
Number of holes : 3080
Number of saw cuts : 6192
Total Volume   : 40.3118088345266 m3
-
E N D   O F   L I S T

```

11.2.5. Conclusion. The integrated design approach allowed us to obtain the production data of every single constructional element. Further, quantitative information about the work steps needed to produce the

elements as well the quantity of raw material used were able to be obtained. In addition to this, an addressing system assigning unique IDs to each element has been defined.

This project allowed us to employ practically the methods elaborated within the chapter 6. By the present project, the notions studied within the scope of this work turned out to be general enough in order to be applied to a real world scenario.

Although we were able to condition the data set describing the constructional elements in the way we wanted, the project has not been executed. The price offers returned by the contractors turned out to be out of the budgetary range of the awarding authority, such that the project had to be much simplified and, finally, it is about to be built without a double curved roof structure.

11.3. Applications to Integrated Production

The findings related to questions of integrated manufacturing have allowed the realization of several small projects. Two examples [11.3.2,11.3.1] show the application of the developed methods used for machining of quad panel structures presented in section 8.4.

We have chosen the present two designs, which are built out of quadrilateral chamfered timber panels. They are highly similar to the timber panel prototypes (cf. 8.4). The post-processor software developed for the production of iterative geometric design was directly applicable for the production of the G-code used to machine the components of the two examples discussed below (cf. 11.3.1 and 11.3.2).

Further, two examples of outsourced G-code post-processors shall be mentioned. While the post-processor written for the 5-axis MAKRA machine is limited to specific part types, the acquired knowledge led to the creation of two post-processors for small 3-axis CNC-routers (cf. 11.3.3). The problem of “programming” 3-axis CNC-machines, which is far less complicated¹, allowed the implementation of more general versions of the G-code post-processors.

11.3.1. Festival Room Model. This project regards 5-axis machining of parts for a reduced scale model of a master student’s design project. Arnaud Bovet designed in the Atelier Weinand design studio 2009 a proposal for a festival room. His proposal was a free-form object built out of 512 quadrilateral timber panels. The CAM solutions existing in our

¹having less undefined parameters

school would have demanded to “program” each part individually, which was impossible within the given time range.

For the production of the model’s components, the methods developed in chapter 7 have been applied on the geometry data provided by the student. Figure 11.3.1 shows the component’s layout for integrated manufacturing. The automatically generated tool-paths are shown in red. The figure shows one of the five raw material boards used for the production of the model’s parts.

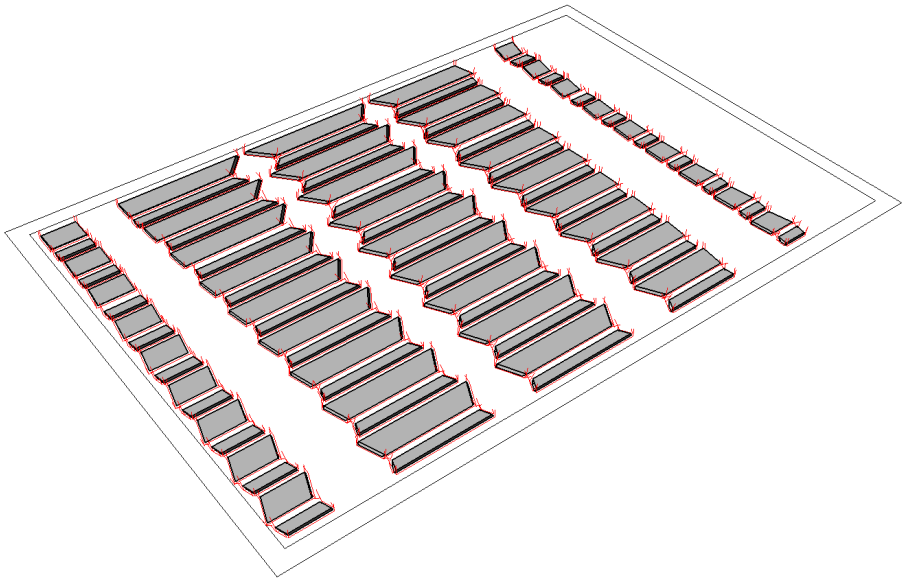


FIGURE 11.3.1. Generated tool-paths (shown in red) and the components (shown in grey)

The fabrication parameters for this project were defined by the chosen tool and the material. A flat tipped 6mm finger tool was used to mill the parts out of a 8mm thick MDF-board. The production of the 512 pieces could be done within one day - five hours of milling. The assembly of the parts is shown by figure 11.3.2.

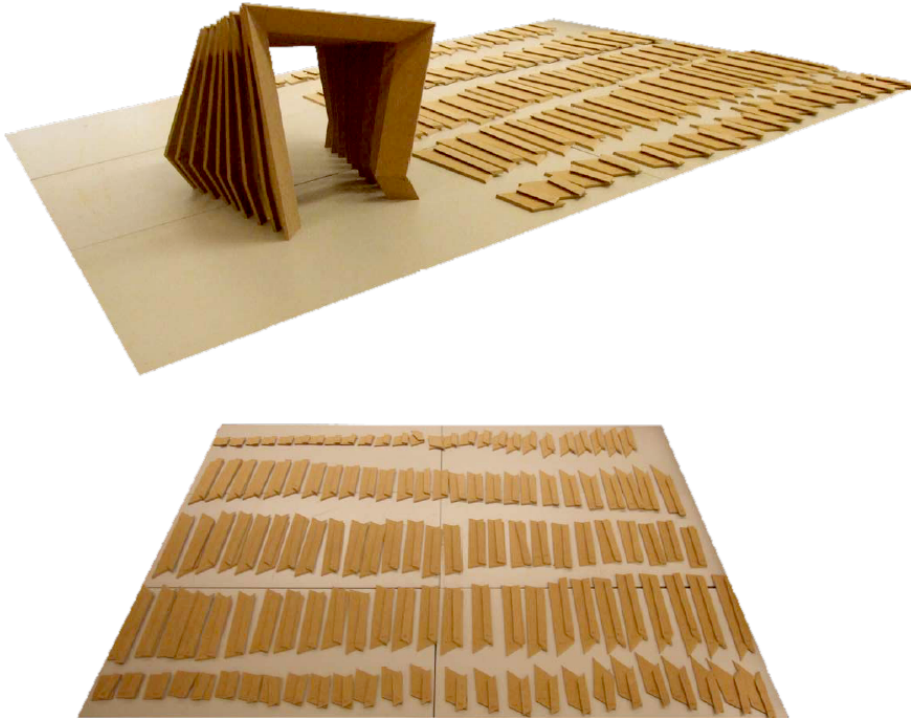


FIGURE 11.3.2. Assembly of the student's model

Note: Although, the project hasn't been geometrically defined by iterative geometric design, fabrication methods developed within the scope of this thesis were able to be applied. The method developed did fit smoothly into digital production chain of this project.

11.3.2. Production of an Origami Eye-catcher. This section presents the production process of an eye-catcher sculpture for the exposition Architexto, which was held in may 2009 in the city of Liège, Belgium. Designed by Hani Buri, the eye-catcher presents a four meter tall folded timber board structure. Analogous to the previous example (cf. section 11.3.1), integrated manufacturing of the components used the methods established in chapter 7.

The global design was composed of 78 pieces. In addition to the contour cut of the parts, holes for dowel connections were machined on the sides of the panels. Further, holes for fixing the parts on the machining table have been generated in an automated manner.

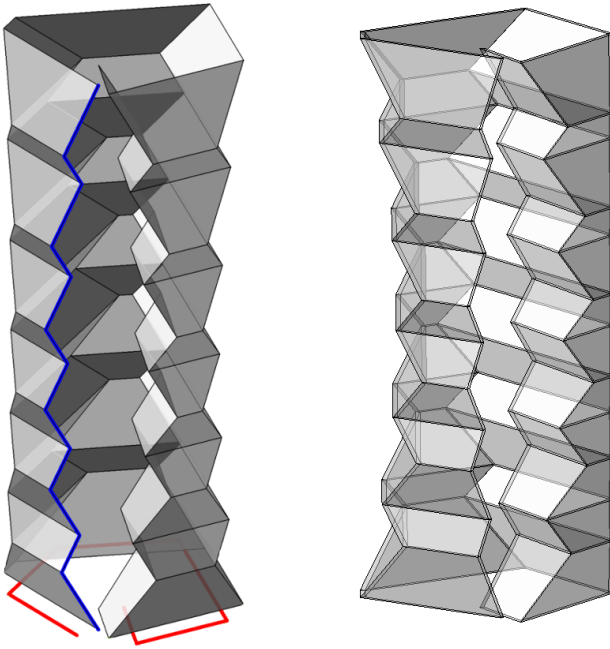


FIGURE 11.3.3. Left: Initial surface; Right: Thickened components

Figure 11.3.3 shows the initial surface designed by Hani Buri. The form-finding parameters have been limited to two curves (shown respectively in blue and in red). To obtain the geometry data of the components, thickening methods of chapter 6.3 have been applied. Furthermore, two different border condition were defined. The top and bottom elements have been cut off horizontally with regards to the fact that the eye-catcher was supposed to be put on a flat floor. The left and the right sides have been cut by right angles. The differentiation of the border conditions was further used for the automated detail generation. Holes for dowel connections were machined in all sides of the center panels. For border and corner panels, this detail was only applied on respectively three and two corresponding sides.

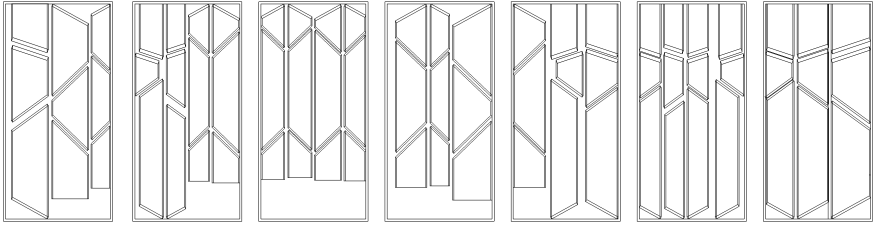


FIGURE 11.3.4. Mapped and packed parts

Figure 11.3.4 shows the constructional components mapped and packed onto the raw material plywood boards. Nota bene that the developable surface design of Hani Buri is very beneficial for efficient packing, which provides densely packed boards with very little material waste. Therefore, the 78 parts could be contained by seven plywood boards of 125cm by 250cm.

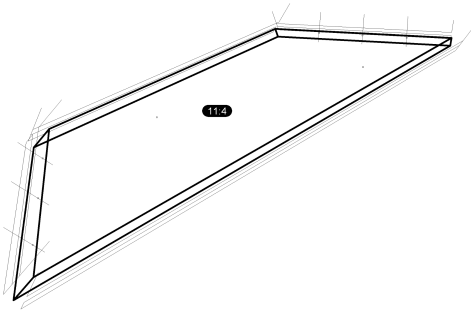


FIGURE 11.3.5. Production of the eye-catcher's parts

The generation of the G-code used several principles described in chapter 7. The contour cut used for milling the global shape of the components was done in three passes (depending on the cutting angle and the material thickness, which was 21mm). The machining of the detail for the dowel fasteners was done by a 6mm piercing drill. The tool-paths of one piece and the machined parts are shown by figure 11.3.5.



FIGURE 11.3.6. Finished eye-catcher

Two photographs of the finished eye-catcher are shown by figure 11.3.6. Besides of its success and its sculptural qualities, this small sized project allowed us to apply several of the methods worked out within the scope of this work. The applications range from data structuring and addressing over thickening and detailing, to mapping, and finally automated G-code creation. They cover a wide spectrum of questions addressed in part II of the present work. Even though this project was relatively small in scale, it was tremendously interesting to our work allowing to test and to verify several elaborated methods at the same time.

11.3.3. 3-axis Machine Post-Processors. Within the scope of this thesis, methods for automatic G-code generation have been developed (cf. 7). With regards to architectural applications and with regards to the machine at the laboratory's disposal, the method has been optimized for a 5-axis milling machine, namely the MAKAMM7s. The machine instructions are based on standardized G-code commands specified in the ISO standard 6983-1:1982. The developed post-processor has been used for the production of several timber plate structures (cf. 8.4, 11.3.1, 11.3.2). For the production of the decor panels presented in section 8.1 and the components of the B-spline shell (cf. section 8.3), a 3-axis CNC router from Zünd was used. The instructions of the controller of this router does not support G-code but extended HP-GL. HP-GL is a graphic language developed by Hewlett Packard for driving ink plotters. The basic command-set might be explained by four commands. Typically, a instruction set of a HP-GL file might look like this:

PU0,0;PD100,100

Literally translated the instruction set shown above says:

PU	Pen up
0,0	Go to position x=0, y=0
;	is the separator
PD	Pen down (starts drawing)
100,100	Go to position x=100, y=100 (linear interpolation)

For driving 2.5D-routers, these two instructions and the PA/PR² commands are the most used of the HP-GL language: plot and coordinates. This corresponds basically to the G1 and the G0 commands of the ISO standard 6983-1:1982. This set of four commands is practically sufficient to machine any polygonal 2D-objects. In addition to straight linear interpolation, CA (circular arc) and CI (circle) commands offer circular interpolation methods.

More detailed information about HP-GL can be found in [Dij99]. While the HP-GL language was initially developed for printers, Zünd extended it for the used of 3-axis machines. Therefore, additional commands defining tool parameters and Z-axis coordinates have been added. The speed and velocity related parameters correspond to the G-code's

²PA = PenAbsolute; PR = PenRelative

feed rate (F-commands). Finally, HP-GL allows working with up to 8 tools, which can be defined by the command SP³.

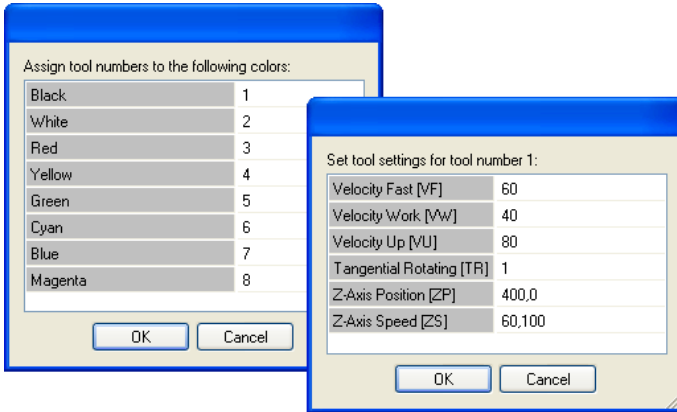


FIGURE 11.3.7. Parameter input dialog

Figure 11.3.7 show the input dialog allowing to set the tool's parameters. Hereby, it is possible to assign different tools to different colors of the drawing. This may allow e.g. to engrave the red curves of a drawing, to cut the blue one and to pen plot the black ones. Multiple tool paths and parameters can be stored within one file.

Below, a extract of a HP-GL-file is shown. It contains a set of instructions for machining 2D-polygons by a Zünd router. The beginning of the file contains a header initializing the machine. Further, tool one is selected (SP1) and its parameters are defined, followed by a series of coordinates. Finally the file ends by the selection of tool zero (SP0⁴) and a pen up command (PU).

³SP = SelectPen. While printers used to work with pens as only tools the command "Select Pen" has been chosen. However, the Zünd controller uses the SP-command to select the tool.

⁴SP0 means that the previously selected too is deselected, since tool 0 is not defined.

IN;PU;
CT1;
SP1;PU;
VF60;
VW40;
VU80;
TR1;
ZP400,0;
ZS60,100;
PA3900,2500;
PD;
PA3538.5,2139.53;
PA3539.14,2817.82;
PA3011.44,2818.32;
(...)
PA3862.34,2087.44;
PA3760.54,2209.81;
PA3971.15,2190.58;
PA3986.11,2354.44;
PA3829.18,2446.29;
PA3900,2500;
PU;
SPO;PU;

The developed Zünd post-processor was only used for the two decor panels presented in section 8.1 and for the production of the components of the B-spline shell (cf. section 8.3). This can be explained by the fact that a lot of third party software exists for the creation of machine instructions for 3-axis CNC-routers. Further, it seems that the manufacturer Zünd is developing a universal printer driver for his machines, which would be able to command the router out of any CAD-software having printing capabilities.

However, a fork project for Haase CNC-routers has been developed. It allows the creation of HP-GL and G-code files from 2d-curve objects. Actually, this post-processor is in daily use for the manufacturing of prototypes by the firm Cute Cut.

11.4. Load Bearing Capacities of Iteratively Designed Geometries

The possibility of the iterative geometric design, that allows the generation of rough and smooth objects, could be employed for the design of bearing plate structures, using the rigidity of the fold to improve the structural strength of free-form surfaces.

Figure 11.4.1 illustrates a preliminary investigation on the bearing potential of such folded structures. This investigation considers the load bearing capacities of free-form designs which are similar to the ones studied in 8.4. The analysis has been carried out by Dipl. Ing. Johannes Natterer, IBOIS-EPFL.

On the left side of figure 11.4.1, a smooth symmetrical plate structure is shown. On its right, the same model with added folds is shown. In order to get a rough idea of the global structural behavior we imagined a plate structure spanning over twelve meters, built of 20mm plywood panels (spruce). First, we designed the smooth dome-like shell structure. Hereon, asymmetrical load of $F_z = 1500\text{N/m}^2$ was applied, which is a typical design value for snow load in central Switzerland. The maximum deflection occurred along the z-axis, which was about 266mm. After adding folds, the FEM-analysis showed a maximum deflection value of 15mm.

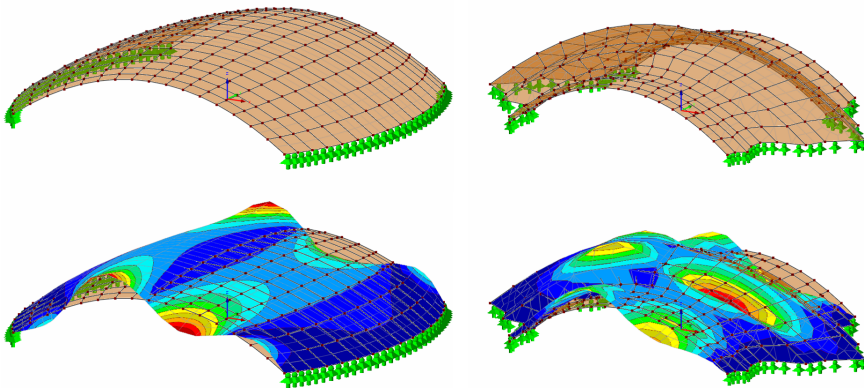


FIGURE 11.4.1. Comparison of the a smooth and rough IFS-surface under asymmetrical load

Note: This analysis was assuming hinge joints between the individual constructional elements. In reality, the joints may present a certain bending rigidity. Since the constructional elements are fixed along all four sides (except the border panels) any possible rotation is greatly limited

by the system. For real scale applications, we think that adequate detailing is highly important. Some questions related to detailing of folded timber plate structures have been addressed in chapter 6.4. Further information about the joints of folded timber plate structures can be found in [BW09] and [HW08a]. Buri et al. [BW09] used 2mm folded steel plates for joining massiv block timber panels of 40mm. Haasis et al. studied in [HW08a] the bending rigidity of screwed connections.

The above mentioned thoughts about detailing and structural properties of the presented work are possible subjects for future investigations.

11.5. Software Implementation

The methods presented in this thesis have intentionally not been related to particular software implementations. The principles of the methods studied were explained in a general way independent from a specific development environment (programming language and platform). Nonetheless, the methods have led to a series of software for design and production of architectural free-form objects. Within the following, the individual software implementations are listed:

- (1) FracMod: CAGD-software based on iterative geometric design. Written in C++ and OpenGL. Aspects: Subdivision point modeling, point weight modeling, projected vector sum modeling.
- (2) SubdivScheme Editor and MultiResMesh Editor: CAGD-software based on iterative geometric design. Written in C++ OpenGL using QT and SimpleViewer. Aspects: Declarative subdivision scheme editing, multi-resolution modeling, planarity optimization.
- (3) CAD-scripts for: Addressing, computation of constructional components, detailing, mapping, packing, tool path generation and post processing. Written in VisualBasic and .Net for Rhinoceros3D.
- (4) Further, empirical testing for the development of the geometric design method has been done in MatLab and AutoCad.

Even though some tools have been implemented for specific commercial products using their own specific commands and languages, the methods worked out in this thesis are general enough to be implemented on other platforms.

Curriculum Vitae

Ivo Stotz
Dipl. Arch. EPFL
Birth date: 27.11.1977
Origin: Basel, Switzerland

Curriculum Vitae:

2007	Foundation of Cute Cut (custom integrated production for everybody)
2005	PhD. Candidate EPFL/ENAC/IS/IBOIS
2004	Employee of Alain Wolff Architectes / Lausanne
2004	Dipl. arch EPFL (Master of Architecture)
2000	Internship at P.Oswalt & K.Overmeyer / Berlin
1999	Internship at Jeroch Kaiser & Partner / Berlin
1998	Civil service
1997	Snowboard teacher
1996	Matura / little Latinum
1977	Born in Basel / Switzerland

Academic Missions:

- PhD. candidate working on the interdisciplinary research entitled “Fractal Geometry and its Applications in the Field of Construction”
- Teaching at the “Atelier Weinand” architectural design studio
- Seminars on “Digital Design and Production”, “CAD-scripting” and “Architectural Geometry”
- Teaching faculty of the course “Timber Construction for Architects and Civil Engineers”
- Thesis assistant for the Master of Architecture, SAR-EPFL

List of Publications:

- I. Stotz, G. Gouaty and Y. Weinand - *Iterative Geometric Design for Architecture* - Journal of the International Association for Shell and Spatial Structures, Vol. 50, Nr. 1, pp. 11-20, 2009.
- G. Gouaty, E. Tosan, I. Stotz and Y. Weinand - *Un modèle itératif de surfaces pour la construction en bois* - Revue Électronique Francophone d'Informatique Graphique, Vol. 3, Nr. 1, pp. 1-12, 2009.
- I. Stotz, G. Gouaty and Y. Weinand - *Iterative Flächengestaltung* - Bulletin Holzforschung Schweiz, Nr. 1, pp. 29-34, 2008.
- G. Gouaty, E. Tosan, I. Stotz and Y. Weinand - *Modélisation de formes fractales basées sur le modèle BCIFS* - JIG08, 2008.
- I. Stotz, G. Gouaty and Y. Weinand - *IFS-Modeling for Feasible Freeform Timber Constructions* - 10th WCTE, 2008.
- I. Stotz, Y. Weinand and G. Gouaty - *Iterative surface design for constructions based on timber panels* - SAH Statusseminar, 2008.
- G. Gouaty, E. Tosan, I. Stotz and Y. Weinand - *Un modèle itératif de surface pour la construction en bois* - GTMG08, 2008.
- Y. Weinand, H. Buri and I. Stotz - *Zukunftsperspektiven aus der Forschung - Praktische Anwendung von Massivholzplatten* - pp. 255-266, 2007.
- E. Tosan, I. Stotz, G. Gouaty and Y. Weinand - *Modélisation itérative de courbes et surfaces: aspect multi-résolution* - GTMG07, 2007.
- E. Tosan, I. Bailly-Salins, G. Gouaty, I. Stotz, P. Buser and Y. Weinand - *Une modélisation géométrique itérative basée sur les automates* - GTMG06, 2006.