2016 CAADence in architecture

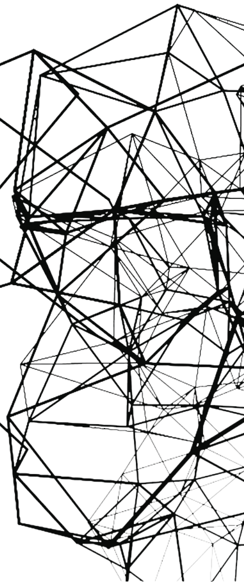
Back to command

Edited by Mihály Szoboszlai

2016 CAADence in architecture

Back to command

Edited by Mihály Szoboszlai



Editor

Mihály Szoboszlai
Faculty of Architecture
Budapest University of Technology and Economics

2nd edition, July 2016

CAADence in Architecture – Proceedings of the International Conference on Computer Aided Architectural Design, Budapest, Hungary, 16th-17th June 2016. Edited by Mihály Szoboszlai, Department of Architectural Representation, Faculty of Architecture, Budapest University of Technology and Economics

Cover page: Faraway Design Kft.

Layout, typography: based on proceedings series of eCAADe conferences

DTP: Tamás Rumi

ISBN: 978-963-313-225-8
ISBN: 978-963-313-237-1 (online version)

CAADence in Architecture. Back to command
Budapesti Műszaki és Gazdaságtudományi Egyetem

Publisher: Faculty of Architecture, Budapest University of Technology and Economics

CAADence in Architecture

Back to command

Proceedings of the International Conference on
Computer Aided Architectural Design

16-17 June 2016
Budapest, Hungary
Faculty of Architecture
Budapest University of Technology and Economics

Edited by
Mihály Szoboszlai

Theme

CAADence in Architecture
Back to command

The aim of these workshops and conference is to help transfer and spread newly appearing design technologies, educational methods and digital modelling supported by information technology in architecture. By organizing a workshop with a conference, we would like to close the distance between practice and theory.
Architects who keep up with the new design demanded by the building industry will remain at the forefront of the design process in our IT-based world. Being familiar with the tools available for simulations and early phase models will enable architects to lead the process. We can get "back to command".
Our slogan "Back to Command" contains another message. In the expanding world of IT applications, one must be able to change preliminary models readily by using different parameters and scripts. These approaches bring back the feeling of command-oriented systems, although with much greater effectiveness.

Why CAADence in architecture?
"The cadence is perhaps one of the most unusual elements of classical music, an indispensable addition to an orchestra-accompanied concerto that, though ubiquitous, can take a wide variety of forms. By definition, a cadence is a solo that precedes a closing formula, in which the soloist plays a series of personally selected or invented musical phrases, interspersed with previously played themes – in short, a free ground for virtuosic improvisation."
Nowadays sophisticated CAAD (Computer Aided Architectural Design) applications might operate in the hand of architects like instruments in the hand of musicians. We have used the word association cadence/caadence as a sort of word play to make this event even more memorable.

Mihály Szoboszlai
Chair of the Organizing Committee

Sponsors

GRAPHISOFT
ARCHICAD

AUTODESK

STUDIO IN-EX
ARCHITECTS & ENGINEERS

MŰEGYETEM 1782

Építészeti Ábrázolás Tanszék
Department of Architectural Representation

Acknowledgement

We would like to express our sincere thanks to all of the authors, reviewers, session chairs, and plenary speakers. We also wish say thank you to the workshop organizers, who brought practice to theory closer together.

This conference was supported by our sponsors: GRAPHISOFT, AUTODESK, and STUDIO IN-EX. Additionally, the Faculty of Architecture at Budapest University of Technology and Economics provided support through its “Future Fund” (Jövő Alap), helping to bring internationally recognized speakers to this conference.

Members of our local organizing team have supported this event with their special contribution – namely, their hard work in preparing and managing this conference.

Mihály Szoboszlai
Chair of the Organizing Committee

Local conference staff

Ádám Tamás Kovács, Bodó Bánáti, Imre Batta, Bálint Csabay, Benedek Gászpor, Alexandra Gööz, Péter Kaknics, András Zsolt Kovács, Erzsébet Kőnigné Tóth, Bence Krajnyák, Levente Lajtos, Pál Ledneczki, Mark Searle, Béla Marsal, Albert Máté, Boldizsár Medvey, Johanna Pék, Gábor Rátonyi, László Strommer, Zsanett Takács, Péter Zsigmond

Workshop tutors

Algorithmic Design through BIM

Erik Havadi

Laura Baróthy

Working with BIM Analyses

Balázs Molnár

Máté Csócsics

Zsolt Oláh

OPEN BIM

Ákos Rechtorisz

Tamás Erős

GDL in Daily Work

Gergely Fehér

Dominika Bobály

Gergely Hári

James Badcock

List of Reviewers

Abdelmohsen, Sherif - Egypt
Achten, Henri - Czech Republic
Agkathidis, Asterios - United Kingdom
Asanowicz, Aleksander - Poland
Bhatt, Anand - India
Braumann, Johannes - Austria
Celani, Gabriela - Brazil
Cerovsek, Tomo - Slovenia
Chaszar, Andre - Netherlands
Chronis, Angelos - Spain
Dokonal, Wolfgang - Austria
Estévez, Alberto T. - Spain
Fricker, Pia - Switzerland
Herr, Christiane M. - China
Hoffmann, Miklós - Hungary
Juhász, Imre - Hungary
Jutraz, Anja - Slovenia
Kieferle, Joachim B. - Germany
Klinc, Robert - Slovenia
Koch, Volker - Germany
Kolarevic, Branko - Canada
König, Reinhard - Switzerland
Krakhofer, Stefan - Hong Kong
van Leeuwen, Jos - Netherlands
Lomker, Thorsten - United Arab Emirates
Lorenz, Wolfgang - Austria
Loveridge, Russell - Switzerland
Mark, Earl - United States
Molnár, Emil - Hungary
Mueller, Volker - United States
Németh, László - Hungary
Nourian, Pirouz - Netherlands
Oxman, Rivka - Israel
Parlac, Vera - Canada
Quintus, Alex - United Arab Emirates
Searle, Mark - Hungary
Szoboszlai, Mihály - Hungary
Tuncer, Bige - Singapore
Verbeke, Johan - Belgium
Vermillion, Joshua - United States
Watanabe, Shun - Japan
Wojtowicz, Jerzy - Poland
Wurzer, Gabriel - Austria
Yamu, Claudia - Netherlands

Contents

Keynote speakers

REINHARD KÖNIG

Reinhard König studied architecture and urban planning. He completed his PhD thesis in 2009 at the University of Karlsruhe. Dr. König has worked as a research assistant and appointed Interim Professor of the Chair for Computer Science in Architecture at Bauhaus-University Weimar. He heads research projects on the complexity of urban systems and societies, the understanding of cities by means of agent based models and cellular automata as well as the development of evolutionary design methods. From 2013 Reinhard König works at the Chair of Information Architecture, ETH Zurich. In 2014 Dr. König was guest professor at the Technical University Munich. His current research interests are applicability of multi-criteria optimisation techniques for design problems and the development of computational analysis methods for spatial configurations. Results from these research activities are transferred into planning software of the company DecodingSpaces. From 2015 Dr. König heads the Junior-Professorship for Computational Architecture at Bauhaus-University Weimar, and acts as Co-PI at the Future Cities Lab in Singapore, where he focus on Cognitive Design Computing.
Main research project: Planning Synthesis & Computational Planning Group see also the project description: Computational Planning Synthesis and his external research web site: Computational Planning Science

BRANKO KOLAREVIC

Branko Kolarevic is a Professor of Architecture at the University of Calgary Faculty of Environmental Design, where he also holds the Chair in Integrated Design and co-directs the Laboratory for Integrative Design (LID). He has taught architecture at several universities in North America and Asia and has lectured worldwide on the use of digital technologies in design and production. He has authored, edited or co-edited several books, including “Building Dynamics: Exploring Architecture of Change” (with Vera Parlac), “Manufacturing Material Effects” (with Kevin Klinger), “Performative Architecture” (with Ali Malkawi) and “Architecture in the Digital Age.” He is a past president of the Association for Computer Aided Design in Architecture (ACADIA), past president of the Canadian Architectural Certification Board (CACB), and was recently elected future president of the Association of Collegiate Schools of Architecture (ACSA). He is a recipient of the ACADIA Award for Innovative Research in 2007 and ACADIA Society Award of Excellence in 2015. He holds doctoral and master’s degrees in design from Harvard University and a diploma engineer in architecture degree from the University of Belgrade.

Changing Tangent and Curvature Data of B-splines via Knot Manipulation

Szilvia B.-S. Béla[1], *Márta Szilvási-Nagy*[2]

[1,2]*Department of Geometry, Mathematical Institute,*
Budapest University of Technology and Economics, Budapest, Hungary
e-mail: {belus|szilvasi}@math.bme.hu

Abstract: *Modifications of B-spline knot values change the parametrization and influence the shape of B-spline curves. Via these computations one can modify B-spline data (derivative, curvature value at a curve point, some points of the control polygon, etc.) such that the new parametrization of the curve satisfies special input conditions of a B-spline algorithm. We give a detailed analysis of operations on knot vectors determining the parametrization of non-uniform B-spline functions. Different knot manipulation techniques are presented using blossoming approach. We describe a new knot manipulation strategy: repositioning of a knot, which is computed directly without knot insertion and removal. This strategy can be used for clamping the control polygon of B-spline curves. As further applications of the knot manipulation we show two methods which modify the tangent and the curvature data in the starting and end points of B-spline curves. These computations are illustrated with nice examples.*

Keywords: *B-spline curves, knot manipulation, end conditions*

DOI: *10.3311/CAADence.1615*

INTRODUCTION

Knot manipulation techniques are widely used to modify the parametrization of B-spline curves. These parameter-transformations are necessary to fulfill geometric constrains in certain points/edges of B-spline curves or surfaces. Such constrains can arise from various user specified input conditions or from the geometry of the model in curve and surface design.

The most important knot manipulation techniques are the knot insertion and removal. These algorithms can be used for degree manipulation, refinement of the knot sequence, changing the contact order of spline segments by raising the multiplicity of knots, clamping or unclamping the control polygon of the curve etc. Formerly several papers have been presented to analyze knot insertion and removal strategies (see [4, 5]). A good survey can be found in the books [6, 7]. Eck et al. [9] also presented a paper which analyses in details the knot removal. As an application of techniques keeping the shape of the input curve the clamping of control polygons is described by Hu et al. [10], which is a special case of the knot modification. Clamping the control polygon of the B-spline is a knot modification which pulls all knot values into one in the end of the knot vector. A further application of knot manipulation is shown in [11], where the authors present a curve merging method with adjusting the knot vectors of the input curves.

The effect of changing one knot in the knot vector and keeping the control polygon unchanged was comprehensively studied by Juhász and Hoffmann [12]. Since this knot manipulation changes the shape of the input curve, the authors applied the technique in different shape control problems [13]. We present here a collection of different knot manipulation techniques which keeps or approximates the shape of the input curve. The insertion and removal techniques are combined in order to perturb a knot in the inner part and in the end of the knot vector. We describe the effect of these knot perturbations on the shape of the B-spline curves. In order to apply these knot manipulations we show how to set the tangent and the curvature values at the endpoints of a B-spline curve and how to generalize this technique to B-spline surfaces.

B-SPLINE CURVES AND KNOT MANIPULATIONS

The Definition of B-spline Curves

A curve $\mathbf{b}(t)$ is called a B-spline of order k, defined on the knot vector $\mathbf{t}=(t_1, t_2, \ldots, t_n)$ where $t_i \leq t_{i+1}$ for all i, if

$$\mathbf{b}(t) = \sum_{i=1}^{n-k} \mathbf{c}_i\, N_i^k(t), \qquad (1)$$

where $\mathbf{C}=(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{n-k})$ are the control points of the curve and $N_i^k(t)$ are the basis functions defined by the recurrence:

$$N_i^1(t) = \begin{cases} 1, & t \in [t_i, t_{i+1}) \\ 0, & otherwise \end{cases}$$

$$N_i^k(t) = \frac{t-t_i}{t_{i+k-1}-t_i}\, N_i^{k-1}(t) + \frac{t-t_{i+k}}{t_{i+1}-t_{i+k}}\, N_{i+1}^{k-1}(t)$$

These curves are piecewise polynomial curves of degree $k-1$ over the parameter domain $[t_k, t_{n-k}]$. Each segment of the curve has the parameter range $[t_i, t_{i+1}]$, where $i=k,\ldots, n-k-1$. These segments are joining to each other in the points $\mathbf{b}(t_i)$ with contact order $k-2$, if all t_i knots are different.
If we change the value of a knot t_i, then the basis functions $N_i^k(t), j = 1, \ldots, j + k - 1$ are changed. By adding or removing a knot value we can change our basis to denser or coarser function set, while the curve will have one more or one less control point and curve segment.

Knot Insertion and Removal Algorithms

Knot insertion is a technique, which raises the number of basis functions used in the assignment of the curve. Thus the insertion of a knot can be derived without changing the shape of the curve. We can express the new control points $\mathbf{C}^*=(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{n-k})$ of the curve via a matrix multiplication,

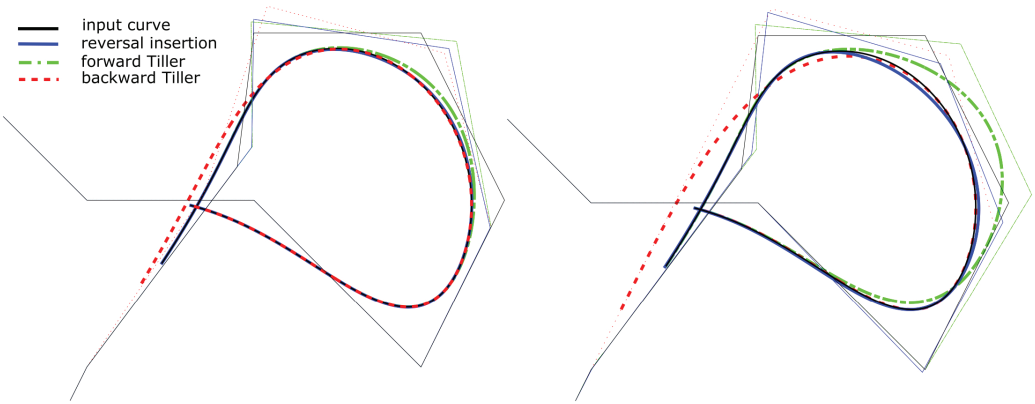
$$\mathbf{C}^* = \mathbf{M}(\mathbf{t}, j; \tau)\mathbf{C}, \qquad (1)$$

where $\mathbf{M}(\mathbf{t}, j; \tau)$ is a bidiagonal matrix and τ is the new knot value inserted to the knot vector $\mathbf{t}$ into the "j+1"th place (see [1] for details).
The removal of a knot from the knot vector results in the basis the reduction of the number of basis functions, thus it cannot be always derived without changing the shape of the curve. Therefore different techniques exist to remove a knot from the knot vector. These techniques generate an approximating curve of the original curve, which keeps the shape of the curve if the removal can be derived without error. The condition when the knot removal does not change the shape of the curve can be found in [9] or in [1] eq. (4). The most common removal techniques are collected in [1]. In the paper three main techniques are considered: the direct inverse method of insertion, and the reversal insertion method proposed by Tiller, which can be computed in two different ways, depending on whether we apply the method forward or backward to the sequence of the control points.

Repositioning of a Knot

Changing one knot value can be understood as consecutive removal of the knot t_j and the insertion of the new perturbed knot value $\tau \in (t_{j-1}, t_{j+1})$. If the knot removal cannot be done without changing the shape of the input curve then we can carry out the knot perturbation using different knot removal strategies. Moreover the order of knot insertion and removal also influences the shape of the output curve. If we apply first the removal then the insertion of a knot, the output curve preserves the shape of the curve generated by the simple knot removal, thus this technique of knot repositioning cannot generate a better approximating output curve as the curve computed by the knot removal (see Figure 1).

Figure 1:
On the left the output curves of consecutive knot insertion and removal, on the right knot insertion after removal are shown on a B-spline curve of degree 4. The insertion after the removal preserves the shape of the curve arisen after the removal the knot.



The recursive computation of the knot insertion and removal can be derived with the help of blossoming technique of B-splines. We can derive similarly the repositioning of a knot as a direct computation on the control points (see [1]). This direct computation technique can be computed in two different ways, too, either forward or backward on the sequence of the control points. The direct repositioning method and the removal after insertion technique have always one of the two computed output curve, which is the same. If the knot t_j is slid to the right to $t_j < \tau$, then the backward computed direct method and the backward computed removal after insertion techniques have the same output curve, if $\tau < t_j$ then the forward computed output curves are the same.

Comparison of the Knot Perturbation Methods
In the following example we compute the output curves of the different knot perturbation algorithms for a B-spline curve of degree 3. We compare here the error occurred in the approximations. The input curve was defined by the control points

$$\boldsymbol{C} = ((0,-1), (-1/\sqrt{2}, -1/\sqrt{2}), (-1,0), (-1/\sqrt{2}, 1/\sqrt{2}), (0,1), (1/\sqrt{2}, 1/\sqrt{2}), (1.5, 1/\sqrt{2}), (1.5,0), (1,0), (1/\sqrt{2}, -1/\sqrt{2}), (0,-1)),$$

on the uniform knot vector t={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}. The knot vector of the input curve (grey curve in Figure 2) was modified such that the knot value "8" was slid to the left to value "7.9". The output curves of the different algorithms are shown in Figure 2. The error of each approximation is computed with piecewise integration on the segments of the curve, and in addition a maximal error value is computed in each segment due to the parameterization. Table 1 shows the error values.

Table 1:
Approximation error measured along each segment of the approximating output curves. For each method the first row shows the total error along the segment, the second row contains the maximal error of the approximation.

Method	[4 , 5]	[5 , 6]	[6, 7]	[7, 7.9]	[7.9,9]	[9,10]	[10,11]
Rev. Ins.	$1{,}46\cdot10^{-5}$	$2{,}96\cdot10^{-4}$	$1{,}06\cdot10^{-3}$	$2{,}29\cdot10^{-3}$	$2{,}10\cdot10^{-3}$	$9{,}59\cdot10^{-4}$	$2{,}78\cdot10^{-4}$
	$7{,}31\cdot10^{-5}$	$4{,}47\cdot10^{-4}$	$2{,}46\cdot10^{-3}$	$4{,}40\cdot10^{-3}$	$4{,}55\cdot10^{-3}$	$2{,}31\cdot10^{-3}$	$4{,}40\cdot10^{-4}$
F. Ins.+Rem.	≈0	≈0	≈0	$3{,}27\cdot10^{-7}$	$1{,}26\cdot10^{-2}$	$5{,}72\cdot10^{-2}$	$2{,}50\cdot10^{-2}$
F. Direct M.	≈0	≈0	≈0	$1{,}60\cdot10^{-5}$	$3{,}93\cdot10^{-2}$	$6{,}41\cdot10^{-2}$	$5{,}21\cdot10^{-2}$
B. Ins.+Rem.	≈0	$3{,}44\cdot10^{-4}$	$8{,}57\cdot10^{-3}$	$1{,}37\cdot10^{-2}$	$9{,}08\cdot10^{-4}$	≈0	≈0
	≈0	$1{,}72\cdot10^{-3}$	$1{,}70\cdot10^{-2}$	$1{,}86\cdot10^{-2}$	$4{,}54\cdot10^{-3}$	≈0	≈0
B. Direct M.	$1{,}02\cdot10^{-3}$	$2{,}62\cdot10^{-2}$	$5{,}79\cdot10^{-2}$	$1{,}07\cdot10^{-2}$	≈0	≈0	≈0
	$5{,}12\cdot10^{-3}$	$5{,}44\cdot10^{-2}$	$6{,}57\cdot10^{-2}$	$3{,}71\cdot10^{-2}$	≈0	≈0	≈0

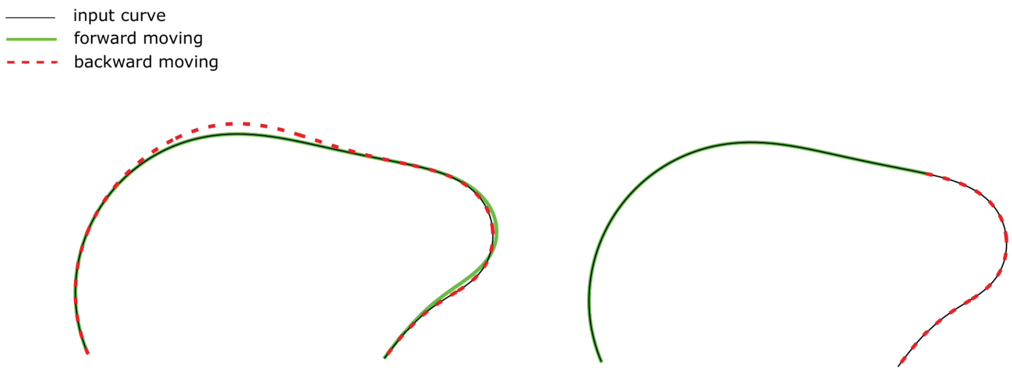


Figure 2:
A B-spline curve of degree 3 is modified. In the left the output curves of the direct method is shown computed forward and backward along the control polygon. In the right the starting segments of the forward computed and the last segments of the backward computed repositioning are shown together.

The error values show us, that the direct computation generates a good approximating curve in that part of the input curve, from which the computations has been started. Figure 2 in the right shows the first four segments of the output curve using the forward computation of direct perturbation and the last three segments of the output curve generated by the backward computed repositioning. The two curves are disjoint in their endpoints associated to the common parameter value 7.9, but both curve segments are preserving better approximation along the first/second half of the input curve, respectively, than other generated output curves.

APPLICATION OF KNOT CHANGING FOR END KNOTS OF THE KNOT VECTOR

Modifying Endknots

If we modify the "endknots" in the knot vector of a B-spline of order *k*, then the recomputation of the first and last *k-1* knots can be always carried out without changing the shape of the curve. The repositioning of the *k*th or *n-k+1*th knots (first and last "important" knots of the curve) cause the extension or shortening of the parameter domain of the curve, thus the starting/endpoint of the curve is moved along the B-spline curve (see Figure 3 a) and b)). If we slide all endknots to the first (last) "important" knot of the curve then we clamp the control polygon to the starting (end)point of the curve, namely the first(last) control point will be moved to the starting (end)point of the curve (Figure 3 c)).

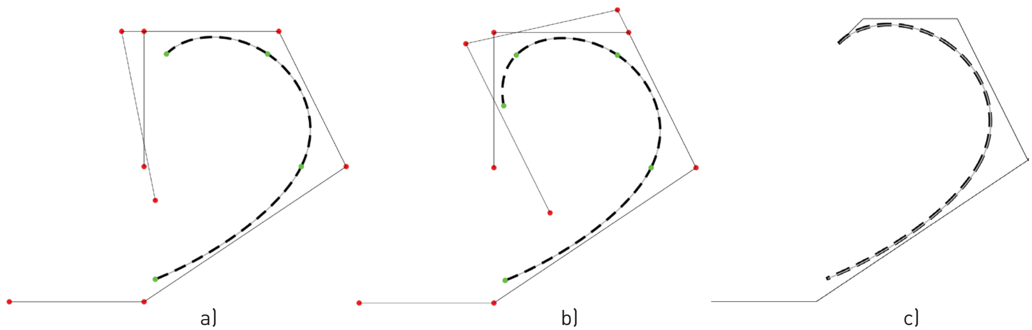
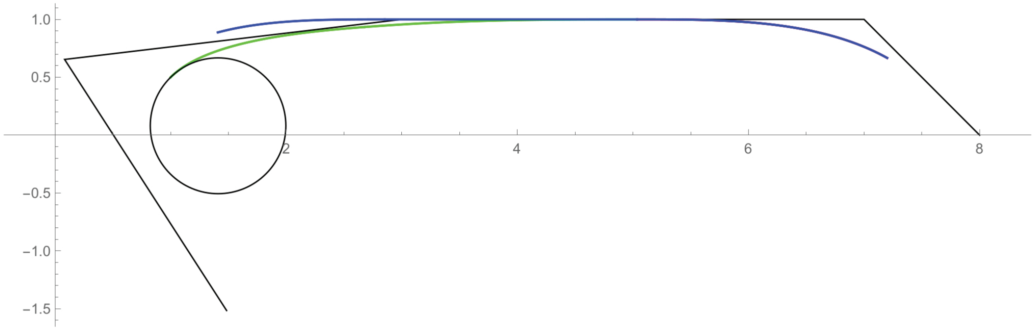


Figure 3:
a-b) Changing third and fourth knots of a cubic curve. The grey curve is the input curve. In the first case the curve is unchanged, in the second case it is extended.
c) Clamped control polygon computed by knot moving.

Figure 4:
The resulting curve shown as a dashed curve, it is determined by the control polygon, the two first control points of which are computed from the given starting point and tangent vector (not shown) with the appropriate knot vector chosen according to the given curvature.



Solution of a second order boundary problem by knot repositioning

In the next example we show a cubic B-spline curve with prescribed second order boundary conditions. As we have shown the length of the *k*th knot interval in the knot vector influences the starting point, the tangent vector at this point and also the shape of the curve. Now we are going to compute the control points of a cubic B-spline curve with given starting point, tangent vector and curvature in this point. The two first control points of the curve are determined by the starting point and the tangent vector at this point, and they are the solution of a system of linear vector equations for each fixed knot vector. The third condition, a prescribed curvature value at this point leads to a non-linear equation, either we want to determine the third control point, or a knot value. In the case of a changing third control point additional conditions would be necessary in order to determine all the coordinates from a scalar equation. Therefore, we have analyzed, how the curvature of a curve of order *k* is depending on the kth knot value perturbed in the fixed *interval (tk-1, tk+1)*. In our case the 4th knot value is changed in the interval determined by the 3th and 5th knot values. We have found that the curvature is monotone decreasing within a bounded interval while the 3th knot interval is growing. Consequently, to each curvature value the corresponding value of the perturbed knot can be determined numerically by a simple interval dividing method.

Figure 5:
The resulting surface has the boundary curve interpolating the given points and tangent vectors. The „longitudinal" isoparametric curves have the prescribed curvature within a relative error bound of 10^{-2}.

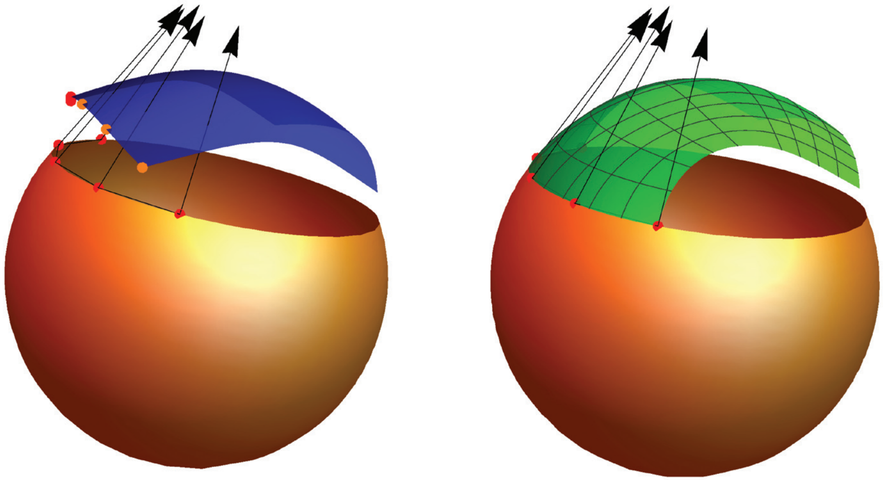


Figure 4 shows the solution of this second order boundary problem for a cubic B-spline curve. The given curvature value is visualized by the osculating circle at the prescribed starting point with a given tangent vector. Of course, the range of this curvature value is limited by the fixed length of the knot interval, where the knot value is moving, but it can be influenced by the length of the tangent vector. This is a subject of our further investigation.
We have extended this method to a bicubic surface. One set of isoparametric curves are computed according to the algorithm developed for cubic B-spline curves. Figure 5 shows a B-spline surface consisting of 2 x 2 patches. The end conditions are visualized on a sphere along a circle. The curvature is the reciprocal value of the radius. In [2] and [3] the end conditions are given by the first and the second derivatives of the curve.

CONCLUSIONS

We have shown new methods for shaping B-spline curves which can be applied to merge and to fit B-spline curves or surfaces. In our algorithms the knot vector determining the basis functions of a B-spline curve has been changed using knot repositioning methods. The different knot perturbation techniques are analyzed and compared via examples. As a possible application it is shown how to set the endpoint, the tangent direction and curvature value of a B-spline curve using knot repositioning in the end of the knot vector. The numerical computations and the figures have been made by Wolfram Mathematica.

ACKNOWLEDGEMENTS

The research of the second author was supported in a cooperation with the Technical University Berlin.

REFERENCES

[1] Béla, Sz. and Szilvási-Nagy, M., Manipulating end conditions of B-spline curves using knot sliding, *Proceedings of Eighth Hungarian Conference on Computer Graphics and Geometry*, Budapest, 2016, p. 83-93.

[2] Szilvási-Nagy, M., Shaping and fairing of tubular B-spline surfaces, *Computer Aided Geometric Design*, vol.14, 1997, p. 699–706.

[3] Szilvási-Nagy, M., Almost curvature continuous fitting of B-spline surfaces, *Journal for Geometry and Graphics*, vol.2, 1998, p.33–43.

[4] Lyche, T. and Morken, K., A data-reduction strategy for splines with applications to the approximation of functions and data, *IMA Journal of Numerical Analysis*, vol.8, 1988, p. 165–208.

[5] Lyche, T. and Morken, K., Knot removal for parametric B-spline curves and surfaces, *Computer Aided Geometric Design*, vol.4, 1987, p. 217–230.

[6] Prautzsch, H., Boehm W. and Plauszny, M., *Bézier and B-spline Techniques*, Springer-Verlag, New York, Inc., USA, 2002.

[7] Goldman, R. N. and Lyche, T., Knot insertion and deletion algorithms for B-Spline curves and surfaces, *Society for Industrial and Applied Mathematics*, 1987.

[8] Tiller, W., Knot-removal algorithms for NURBS curves and surfaces. *Computer-aided Design*, vol.24, 1992, p. 445–453.

[9] Eck, M. and Hadenfeld, J., Knot removal for B-spline curves, *Computer Aided Geometric Design*, vol.12, 1995, p. 259–282.

[10] Hu, S.-M., Tai, C.-L. and Zhang, S.-H., An extension algorithm for B-splines by curve unclamping, *Computer-Aided Design*, vol.34, 2002, p. 415-419.

[11] Tai, C.-L., Hu, S.-M., and Huang, Q.-X., Approximate merging of B-Spline curves via knot adjustment and constrained optimization. *Computer-Aided Design*, vol. 35, 2003, p. 893-899.

[12] Juhász, I. and Hoffmann, M., The Effect of Knot Modifications on the Shape of B-spline Curves, *Journal for Geometry and Graphics*, vol.5, 2001, p. 111-119.

[13] Juhász, I. and Hoffmann, M., Constrained shape modification of cubic B-spline curves by means of knots, *Computer-Aided Design*, vol. 36, 2004, p. 437-445.

Author's index

The aim of these workshops and conference is to help transfer and spread newly appearing design technologies, educational methods and digital modelling supported by information technology in architecture. By organizing a workshop with a conference, we would like to close the distance between practice and theory.

Architects who keep up with the new designs demanded by the building industry will remain at the forefront of the design process in our information-technology based world. Being familiar with the tools available for simulations and early phase models will enable architects to lead the process.
We can get “back to command”.
The other message of our slogan is <Back to command>.
In the expanding world of IT applications there is a need for the ready change of preliminary models by using parameters and scripts. These approaches retrieve the feeling of command-oriented systems, although, with much greater effectiveness.

Why CAADence in architecture?

"The cadence is perhaps one of the most unusual elements of classical music, an indispensable addition to an orchestra-accompanied concerto that, though ubiquitous, can take a wide variety of forms. By definition, a cadence is a solo that precedes a closing formula, in which the soloist plays a series of personally selected or invented musical phrases, interspersed with previously played themes – in short, a free ground for virtuosic improvisation."

CAADence in Architecture
Back to command
International workshop and conference
16-17 June 2016
Budapest University of Technology and Economics
www.caadence.bme.hu

ISBN 978-963-313-225-8
9 789633 132258