LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
FACULTY OF TECHNOLOGY MANAGEMENT
DEPARTMENT OF INFORMATION TECHNOLOGY

NURBS Toolbox for Matlab

The topic of the thesis has accepted October 3rd, 2008.
Examiner: D.Sc. (Tech.) Arto Kaarna

Supervisor: D.Sc. (Tech.) Arto Kaarna

Lappeenranta, December 11th, 2008

Antti Karjalainen
Punkkerikatu 2 A 13
53850 Lappeenranta
Tel. +358 (0)40 511 8979

ABSTRACT

Lappeenranta University of Technology
Faculty of Technology Management
Department of Information Technology

Antti Karjalainen

NURBS toolbox for Matlab

Thesis for the Degree of Bachelor of Science in Technology

2008

32 pages, 8 figures, 6 tables and 2 appendices.

Examiner: D.Sc. (Tech.) Arto Kaarna

Keywords: NURBS, Matlab, toolbox, Mex

NURBS are widely used parametric approximation curves or surfaces. NURBS can be applied to the many applications. Examples of these applications are some computer-aided design applications and some medical applications. Use of NURBS is very intuitive.

The objective of this work was to implement the NURBS toolbox in the Matlab environment. Matlab is a program for many kinds of computational purposes. Matlab is also a programming language. NURBS toolbox implemented in this work offers a user an opportunity to use functions of this toolbox as parts of the user's own programs. Current version of NURBS toolbox includes functions for NURBS curve and surface evaluation. The toolbox is designed such, that it allows extensions and enhancements in the future.

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
Teknistaloudellinen tiedekunta
Tietotekniikan osasto

Antti Karjalainen

NURBS-ohjelmakirjasto Matlab-ympäristöön

Kandidaatintyö

2008

32 sivua, 8 kuvaa, 6 taulukkoa ja 2 liitettä.

Tarkastaja: D.Sc. (Tech.) Arto Kaarna

Hakusanat: NURBS, Matlab, ohjelmakirjasto, Mex

NURBS:t ovat laajasti käytettyjä parametrisia approksimointi käyriä tai pintoja. NURBS:eja voidaan hyödyntää monissa sovelluksissa, näitä sovelluksia ovat esimerkiksi geometrian esittäminen tietokoneella ja lääketieteelliset sovellukset. NURBS:ien käyttö on hyvin intuitiivista.

Tässä työssä tehtiin NURBS-ohjelmakirjasto Matlab-ympäristöön. Matlab on moninaisiin laskentatarkoituksiin käytetty ohjelma ja ohjelmointikieli. Tässä työssä toteutettu ohjelmakirjasto tarjoaa käyttäjälle mahdollisuuden käyttää NURBS:ien laskentaa osana käyttäjän omia ohjelmia. Toteutettu NURBS-ohjelmakirjasto sisältää NURBS:ien laskentaan tarkoitetut funktiot. Ohjelmakirjasto on toteutettu siten, että sitä voidaan laajentaa ja parantaa tulevaisuudessa.

PREFACE

This work was made at Lappeenranta University of Technology in Laboratory of Information Technology in summer and autumn 2008. I want thank Laboratory of Information Technology for good advices. Work atmosphere in the Laboratory of Information Technology was especially good.

Lappeenranta, November 24th, 2008

Antti Karjalainen

CONTENTS

ABBREVIATIONS AND SYMBOLS

`CAD`	Computer-Aided Design
`MEX`	Matlab Executable
`NURBS`	Non-Uniform Rational B-Splines

1 INTRODUCTION

1.1 Background

Non-uniform Rational B-Splines (NURBS) are parametric curves or surfaces. NURBS can be shaped intuitively and with NURBS many kinds of different shapes can be presented. First NURBS have been used in CAD applications [1]. NURBS have been developed from B-splines which are developed from Bezier curves and surfaces [1, 2].

After the first CAD applications NURBS have been used in various applications [1]. Usually NURBS are used to present some shapes. Example of this is modeling of a heart of a human[3]. Sometimes mathematical properties of NURBS are used in simulations. Example of this is modeling of a blood vessels using NURBS and then simulating the blood flow in the vessels [4].

1.2 Objectives and Restrictions

The purpose of this work was to implement NURBS toolbox into Matlab environment. Later on NURBS toolbox refers to the toolbox implemented in this work. It can be used to any kind of tasks where one needs to evaluate NURBS. Maintenance of NURBS toolbox should be easy. So further development of the NURBS toolbox won't be problem. The NURBS toolbox should be efficient and also easy and intuitive to use.

The NURBS toolbox should be able to evaluate NURBS curves and surfaces in 3-dimensional space. User will give the order of NURBS, control points, weights and a knot vector to NURBS toolbox and also parametric points, where NURBS will be evaluated. The NURBS toolbox returns curve and surface values corresponding to the given parametric points. For curves the values are thought of as single points or as a continuous curve consisting of line segments. Evaluation of surfaces is divided into two cases. In the first case single scattered parametric points are evaluated and in the second, a regular net of the parametric points is evaluated.

Calculations will be done using Matlab MEX-functions, which are extensions to Matlab M-functions. Only very basic tools will be implemented. The NURBS toolbox will only evaluate NURBS at the given parametric points. For example evaluations of derivatives of NURBS curves or surfaces will not be implemented, neither knot-insertion or knot-removal algorithms.

In this work we will give examples which show how changing of values of parameters will affect to the shape of a curve or a surface. This will help users of the NURBS toolbox to use it in an efficient way. Users will also learn about NURBS via examples shown. The computation efficiency of the NURBS toolbox will be compared to existing toolboxes.

1.3 Structure of the Thesis

Chapter 2 contains the theory behind NURBS. Some properties of NURBS and few applications of NURBS are presented in Chapter 2.1. In Chapter 2.2 we explain what is Matlab. Some concepts of Matlab are introduced. In Chapter 2.3 we introduce research problem of this work.

In Chapter 3.1 we explain structure of the NURBS toolbox. Some examples about using NURBS curves and surfaces are shown in Chapter 3.2.

In Chapter 4.1 we compare our NURBS toolbox to other toolboxes. In Chapter 4.2 we will tell which parts of the NURBS toolbox could be developed further.

2 BACKROUND

2.1 NURBS

NURBS curves and surfaces are generalizations of B-splines curves and surfaces. Similarly, B-splines curves and surfaces are generalizations of Bezier curves and surfaces. B-spline basis function, used in this work, has published 1972 [5, 6]. Earlier formula for basis function also exists but the one used in this work is better for computer implementations. Earlier formula is published at late 1940's [1]. First applications of B-splines have been a tool for CAD [1]. CAD techniques using B-splines was introduced first at 1973 [1]. Earliest study using NURBS has been published in 1975 [1].

NURBS curves are parametric. Definition of NURBS curve $C(u)$ is

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\omega_i P_i}{\sum_{i=0}^{n} N_{i,p}(u)\omega_i} \qquad a \leq u \leq b \tag{1}$$

where $N_{i,k}(u)$ are basis functions. Definition of $N_{i,k}(u)$ is

$$N_{i,0} = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{2}$$

where u_i are elements of non-uniform and non-periodic knot vector

$$U = \{u_0, \ldots, u_m\} = \{\underbrace{a, \ldots, a}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{b, \ldots, b}_{p+1}\} \tag{3}$$

P_i are the control points, ω_i are the weights and p is the a degree of the polynomial curve. $n+1$ is an amount of control points P_i. $m+1$ is a length of the knot vector U. Between m, n and p holds

$$m = n + p + 1 \tag{4}$$

Order of the curve is $p+1$. Because values of the first $p+1$ elements of the knot vector U are all a NURBS curve begins from the first control point. Because values of the last $p+1$ elements of the knot vector U are all b NURBS curve ends to the last control point. For these definitions see [1, 2, 5, 6, 7, 8, 9, 10, 11].

NURBS surface $S(u,v)$ is defined similarly as a curve using a tensor product

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{k} N_{i,p}(u)N_{j,q}(v)\omega_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{k} N_{i,p}(u)N_{j,q}(v)\omega_{i,j}} \tag{5}$$

$N(u)$ and $N(v)$ are defined in Eq. 2 [1, 2, 7, 8, 9, 10].

Reasons why Bezier curves and surfaces has been developed to B-spline curves surfaces and finally to NURBS curves and surfaces is that NURBS have many nice properties [1, 2]. First important property is that NURBS offer a mathematical form of different kinds of shapes. NURBS can present analytic shapes as well as free-form shapes. An important property for computer applications is that the evaluation of NURBS is computationally stable. By manipulating the knot vector, control points, weights and degree, NURBS can get very many different shapes.

In addition to CAD applications NURBS has been applied to various purposes [1, 2]. First example is from the field of simulation [12]. In this example a geometrical structure of antennas is modeled. These models can be used in simulations of radiation patterns.

Next example is about presenting real world's objects with NURBS [13, 14, 15]. The presentation is done by fitting a NURBS surface to a 3-dimensional data set of points from a surface of a real object. In one study these 3-dimensional surfaces are measured from a 2-dimensional image [13].

NURBS can be used in modeling macromolecular structures [16]. In this case makes NURBS presenting shape of a molecule easier. In another study NURBS are used to represent sets of spectral signatures [17].

NURBS have some applications in the field of medical science [3, 4, 18, 19]. Here are presented few examples. NURBS are used to develop torso phantom of the measured data of a patient [3, 18]. From developed model can doctors examine human body as 3-dimensional model. In another similar study NURBS are used to present skeletal muscular model of a human body [19]. Last example is about modeling living tissue [4]. NURBS are used to model blood vessels. Then these NURBS models can be used in modeling of blood flows in the vessels [4]. An example of a blood vessel model is shown in Figure 1.

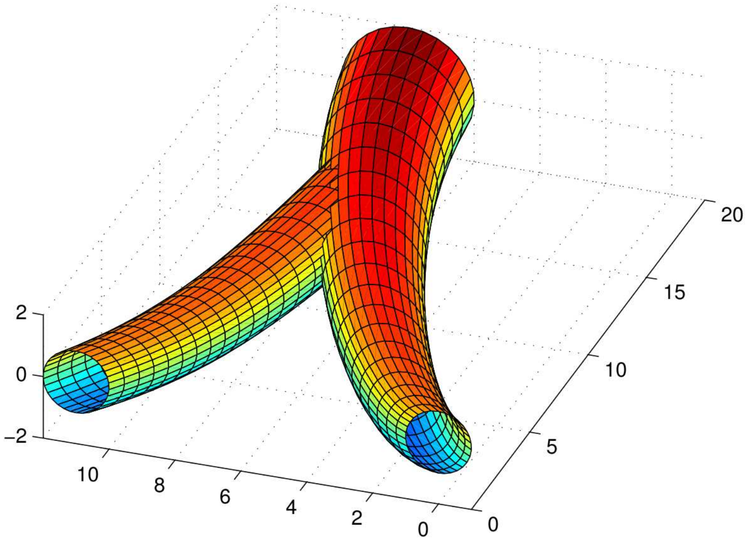

Figure 1. Intersection of blood vessels modeled using NURBS [4].

2.2 Matlab

Matlab is a programmable computation environment. Matlab is used via the command line or via m-files. There are two kinds of m-files. First format of the m-file is a function (M-function). A function in Matlab can take in several input parameters and it can return several output parameters. Second format of the m-file is like the input from the command line. In this case all commands in m-file are executed just like the commands from the command line.

In this work we are using MEX-functions. MEX-functions are used via m-files. In this work MEX-functions are written with C-language. MEX-functions are used because they work more efficiently than m-file functions. NURBS curves or surfaces can be visualized using Matlab. Structure arrays are one way in Matlab to store data. Structure array consists of fields. Every field has a name. Datatypes of fields inside of one structure array can differ from the each others. Cell arrays are multidimensional arrays. Elements of cell array are other arrays.

2.3 Research problem and how to solve it

Matlab is a good environment for many kinds of prototype, testing and research purposes. Matlab environment can be extended by many kinds of toolboxes. Nice properties for toolboxes are efficiency and possibility for easy future extensions and enhancements. The source code of all functions should be well commented. NURBS are interesting mathematical tool for approximation. Due to these facts the research problem of this work is the following:

How efficient and easy to use NURBS toolbox can be implemented to Matlab environment so that easy future extensions and enhancements are possible?

In implementation of NURBS toolbox we will use definitions from Chapter 2.1. We will also modify these algorithms to make the NURBS toolbox work more efficiently. The functions will be implemented such that they are easy to use. We will also provide examples how to use all the properties of NURBS to manipulate NURBS curves or surfaces. To allow easy future extensions and enhancements the source code of all functions in NURBS toolbox will be well commented.

3 PRACTICAL PART

3.1 Structure of NURBS toolbox

Subfunctions

NURBS toolbox works in Matlab environment. The NURBS toolbox consist of functions which can be called in Matlab. The NURBS toolbox consist of four types of files. Current version number of the NURBS toolbox is 2.3. Total number of NURBS toolbox's files is 18. Six of these files are source code files for the MEX-functions. These six source code files are written in C-language. File extension of these files is c. Three of the NURBS toolbox's files are the mex-files. These three files are compiled from six source code files. File extension of these mex-files in Linux is mexglx. Next seven files of the NURBS toolbox are m-files. File extension of m-files is m. One of the m-files is makefile for compiling the mex-files from the source code files. Via other six m-files user uses the NURBS toolbox. Via four of these six m-files the user can in practice use the NURBS toolbox. Two last m-files contain examples. Last two files of whole NURBS toolbox are text files. First of these includes a list of NURBS toolbox's files and the second includes the version history.

User Interface

A user uses the NURBS toolbox via four m-files. Two of these files are for evaluating NURBS curves in 2- or 3-dimensional space. The other two m-files are for evaluating NURBS surfaces. These all four m-files are Matlab functions. The first of NURBS curves evaluation function only performs the evaluation and then returns the results. The second NURBS curves evaluating function first checks if the inputs given by user are valid and then performs the evaluation. Similarly the first surface evaluating file only performs the evaluation and then returns the results and the second one also checks the inputs. None of the four functions user interface functions does not perform the actual calculations. These four functions just get the input values from the user and pass the results to the user. Actual calculations are done in sub functions. In the NURBS toolbox there are three sub functions for the actual evaluation: one for curve evaluation and two for surface evaluation. These three sub functions are in the mex-files. The difference between the two surface evaluation functions is that the first function handles the case where the user wants to evaluate scattered pairs of the parametric points. In the second function the user wants to evaluate a uniform grid of the parametric points.

Example Files

Functions are implemented so that the NURBS toolbox is intuitive to use. Two m-file examples and examples in m-file functions makes using the NURBS toolbox clearer to the user. In addition source code files are well commented so maintenance of the NURBS toolbox is possible.

3.1.1 Function calls

NURBS Curve

Input parameters and output parameter of the two NURBS curve evaluation functions are the same. First input parameter of the NURBS curve functions is a structure array. This structure must include three fields. Names of these fields must be `p`, `U` and `Pw`. `p` is an order of the NURBS curve and its single integer. `U` is a knot vector of the NURBS curve. `U` is a vector of real numbers. `Pw` is a control point matrix of real numbers. Coordinates of one dimension are always at one row in `Pw`. Number of rows in `Pw` depends on the dimension of the NURBS curve. The dimension of the NURBS curve can be two or three. Weights of control points are in the next row after the coordinates. See an example of the input and the output parameters from Appendix A.

The NURBS curve evaluation function can evaluate one or more NURBS curves at the same time. Control points and weights of several NURBS curves must put one after the other to the rows of `Pw`. If evaluating more than one NURBS curve at the same time the order of the knot vector and the dimension of the NURBS curve must be same for every curve. Second input parameter of the NURBS curve function is the dimension of the NURBS curve, it is a single integer. Allowed values for the second input parameter are 2 and 3. Last input parameter of the NURBS curve evaluation function is a vector for the parametric points. Value of the NURBS curve is evaluated at these parametric points. The output parameter of the NURBS curve evaluation function is a matrix. Similarly as in matrix `Pw`, values of each dimension are on own row in the output matrix. The columns of an output matrix are corresponding to the given parametric values. The weights of the control points are not returned. If evaluating several NURBS curves at the same time the output values of the different NURBS curves are placed in the output matrix one after the other. See an example of the input and the output parameters from Appendix A.

NURBS Surface

Input and output parameters of two NURBS surface evaluation functions are similar. The first input parameter of the NURBS surface evalution function is a structure array. This structure must include five fields. Names of these fields must be `p`, `U`, `q`, `V` and `Pw`. `p` is a order of the NURBS surface in the direction of the parameter *u* (see Eq. 3) and it is a single integer. `U` is a knot vector of NURBS surface in the direction of the parameter *u* (see Eq. 3). `U` is a vector of real numbers. `q` is a order of the NURBS surface in the direction of the parameter *v* (see Eq. 3) and it is a single integer. `V` is a knot vector of the NURBS surface in the direction of the parameter *v* (see Eq. 3). `V` is a vector of real numbers. `Pw` is an array of control points as real numbers. The coordinates of one dimension are always at one row in `Pw`. First dimension of `Pw` is for different coordinate components of the NURBS surface. If several surfaces are evaluated at the same time are control points of the different NURBS surfaces placed in the `Pw` one after the other. Second dimension of array `Pw` is for the control points in the direction of the parameter *u* (see Eq. 3). Third dimension of array `Pw` is for the control points in the direction of the parameter *v* (see Eq. 3). If evaluating more than one NURBS surface at the same time orders, knot vectors and dimensions of the NURBS surface must be same for every surface. See an example of the input and the output parameters from Appendix B.

Last input parameter of a NURBS surface can be given as a net or as scattered parametric points. The last input parameter is either a cell array or a matrix. If last input parameter is the cell array it must consist of two vectors. The first vectors presents parametric points in the direction of the parameter *u* (see Eq. 3). Second vectors inside of cell array presents parametric points in the direction of the parameter *v* (see Eq. 3). Together these vectors inside the cell array are presenting a two dimensional net of the control points. The surface is evaluated in knots of this net. If the last input parameter is a matrix the first row of this matrix is presenting parametric points in the direction of the parameter *u* (see Eq. 3) and second row the parametric points in the direction of the parameter *v* (see Eq. 3). Values of the surface are now evaluated in parametic points corresponding to the columns of the parameteric points in the input matrix. Output parameter of the NURBS surface evaluation function is a 3-dimensional array or matrix depending on the evaluation of the net of parametric points or the single scattered points. When evaluating the net of parametric points the output array is similar to the input array `Pw`. When evaluating the single scattered points the output matrix is similar to the output matrix when evaluating NURBS curves. If evaluating several NURBS surfaces at the same time are the output points of the different NURBS surfaces one after the other in the ouput array. The input and output parameters of NURBS curve and surface evaluation functions are commented in the corresponding m-files. These comments can be seen for example via the Matlab's help-function.

3.1.2 Algorithms

Actual programs behind the evaluation of NURBS curves or surfaces are in three MEX-functions. Four user interface m-file functions (see Chapter 3.1) are calling MEX-functions. Programs are divided into six source code files which have been written using C-language. Three of these source files are kind of main programs. M-files are calling these programs. Other three files are sub programs for the three main programs.

`nrbsFindSpans` is a first sub program. The purpose of it is to find all knot spans of the knot vector U. The program simply goes through the given knot vector U and returns the indexes to the points where the knot spans begins and ends. The program returns also the number of the knot spans.

`nrbsFindIndex` is a second sub program. It finds the index to which knot span of U a parametric point u belongs to. Program goes through the knot span indexes given by the `nrbsFindSpans`. The program returns an index to the knot span indexes.

`nrbsBasisFuns` is a third sub program. It computes the nonvanishing basic functions and returns them evaluated using one certain parametric value u [1].

`nrbsCurve` is a first main program. It is for evaluating of NURBS curves. This program gets its inputs from the user via an m-file function which is for evaluating NURBS curves. At first `nrbsCurve` multiplies parametric points by weights. Then the `nrbsCurve` uses sub program `nrbsFindSpans` to get knot spans. Next the parametric points are evaluated for each knot span separately. The second and third sub programs `nrbsFindIndex` and `nrbsBasisFuns` perform the evaluation. If the values of the parametric points in vector u are in increasing order then the program will work faster.

`nrbsSurfaceScatter` is a second main program. It is for evaluating the NURBS surfaces in single scattered parametric points. This program evaluates

`nrbsSurfaceGrid` is a third main program. It is for evaluating the NURBS surfaces at net of parametric points.

In programs `nrbsSurfaceScatter` and `nrbsSurfaceGrid` surface is evaluated at each parametric point one by one. `nrbsSurfaceScatter` and `nrbsSurfaceGrid` uses the following sub programs: `nrbsFindSpans`, `nrbsFindIndex` and `nrbsBasisFuns`.

3.2 Examples of using NURBS toolbox

NURBS curves and surfaces can get different shapes. Evaluation results of NURBS curve or surface can also include discontinuous points. NURBS parameters are control points P_i, weights ω_i, order and knot vector U. These parameters can be used to manipulate the shape of NURBS curves or surfaces.

3.2.1 NURBS curves

The first example of NURBS curves is presented in Figure 2. This curve has ten control points. The control points are presented in the figure. The order of the control points is presented with a dotted line. The curve goes through the first and the last control point (See Chapter 2.1). NURBS curve can also go through the other control points but this is quite rare.

The next example is a modification of first example. Figure 3 presents same curve as Figure 2 except that one control point is modified. P_{8_a} is the original control point in Figure 3 and P_{8_b} and P_{8_c} are the modifications of P_{8_a}. Another observation is that the modification of a control point affects to the curve over three control points, the modified one included. Actually the influence distance of the modification of P_i depends on the order and the knot vector U [1]. It is a good property for an approximation polynomial that a modification of one point is not affecting too far from that point.

In the third example is showed how a change of the order affects to the shape of the curve. This example is presented in Figure 4. Curve becomes stiff when the order increases. When changing the order of the curve, the amount of control points or knot vector must change too (See Eq. 4).

In the fourth example is showed how the manipulation of the knot vector affects to the shape of the curve. The Fourth example is presented in Figure 5. When the knot vector has no double elements the curve is smooth. If two or more elements of the knot vector are

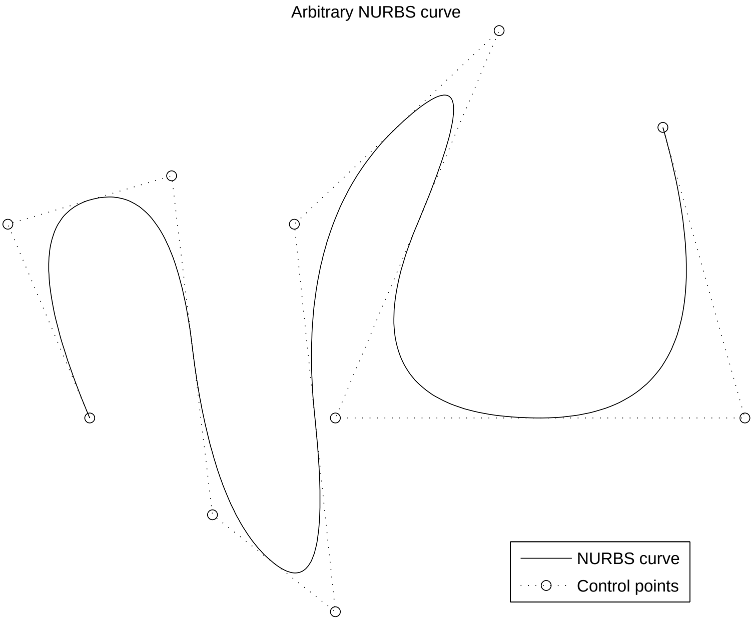

Figure 2. NURBS curve and its ten control points, all weights are ones, knot vector of the curve is $U = \{0,0,0,\frac{1}{8},\frac{2}{8},\frac{3}{8},\frac{4}{8},\frac{5}{8},\frac{6}{8},\frac{7}{8},1,1,1\}$, order is 3.

equal the curve has a discontinuous point. Continuity depends on the order of the curve and the knot vector. Continuity of the curve is C^{p-k} (continuity can be different to the all knot spans) where p is the degree of the curve (See Chapter 2.1) and k is a number of equal elements of the knot vector. In the example four degree of the curve is 2. Because for the control point P_4 $k = 2$ the curve is $C^{2-2} = C^0$ continuous at the P_4 (See Figure 5). Because of the similar reason the curve is discontinuous at the control point P_7 (See Figure 5).

The fifth example presents how the manipulation of the weight of one control point changes the shape of a NURBS curve. This example is presented in Figure 6. Parameters in this example are the same as in the Example 1 except the value of the weight of the control point P_8 has been varied. The Figure 6 shows that when the values of the weight of the control point increases then the curve moves closer to the corresponding control point. Correspondingly, when the weight of the control point decreases the curve moves further away from the corresponding control point.

The sixth example is about creating analytic shapes using NURBS. This example is presented in Figure 7. A circle of this example consists of nine control points. The first and the last control points are at the same point (See the control points P_1 and P_9). The order of this NURBS curve is three. The knot vector of this NURBS curve is

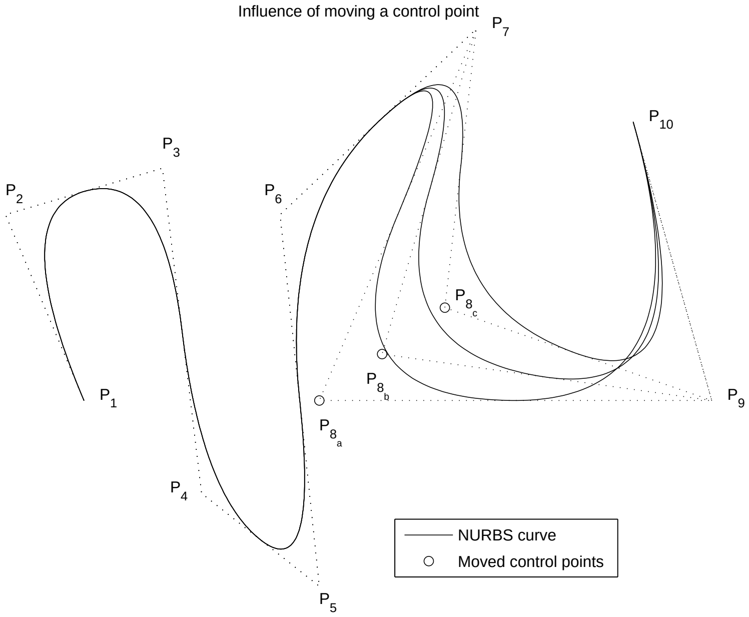

Figure 3. Influence of modification of one control point. All other parameters are the same as in the first example.

$U = \{0,0,0,\frac{1}{4},\frac{1}{4},\frac{2}{4},\frac{2}{4},\frac{3}{4},\frac{3}{4},1,1,1\}$. The weights of the control points are $\omega = \{1,\frac{\sqrt{2}}{2},1,\frac{\sqrt{2}}{2},1,\frac{\sqrt{2}}{2},1,\frac{\sqrt{2}}{2},1\}$.

3.2.2 NURBS surfaces

Handling of NURBS surfaces is quite similar to handling of NURBS curves. The biggest difference is that the NURBS surfaces are computed using two parameters u and v. Both u and v can be thought as of parameters of NURBS curves which are combined to the NURBS surface using a tensor product, see Eq. 5. Because u ja v separately are behaving like NURBS curves all six manipulation methods in Chapter 3.2.1 can be applied also to manipulation of the NURBS surface.

In Figure 8 there is a example of an arbitrary NURBS surface. Some control points of the NURBS surface are visible in Figure 8. The surface has four control points in direction u and five control points in direction v. The orders in both directions are three. The weights for all control points are ones. The knot vector in the direction u is $U = \{0,0,0,\frac{1}{2},1,1,1\}$ and the knot vector in direction v is $V = \{0,0,0,\frac{1}{2},\frac{1}{2},1,1,1\}$.

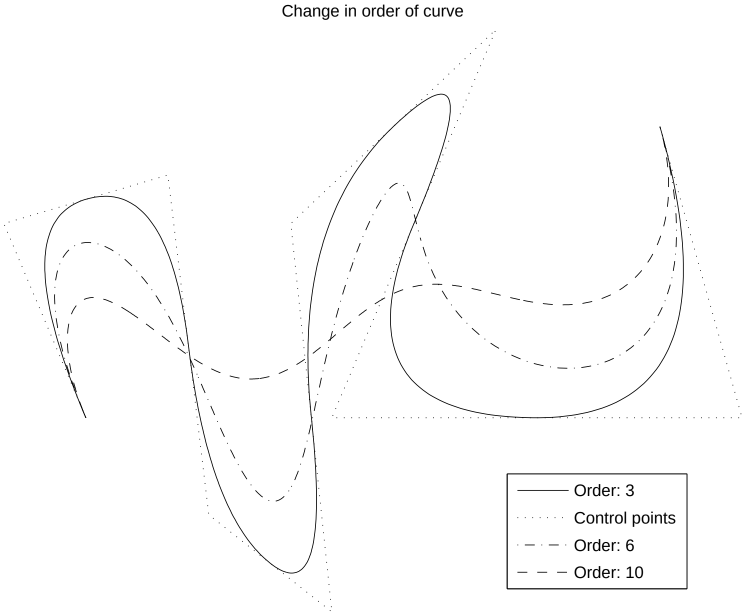

Figure 4. Influence to the shape of curve when changing the order of the curve. For the curve which order is three the knot vector is the same as in Example 1. The knot vector for the curve with order 6: $U = \{0,0,0,0,0,0,\frac{1}{5},\frac{2}{5},\frac{3}{5},\frac{4}{5},1,1,1,1,1,1\}$. The knot vector for the curve with order 10: $U = \{0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1\}$.

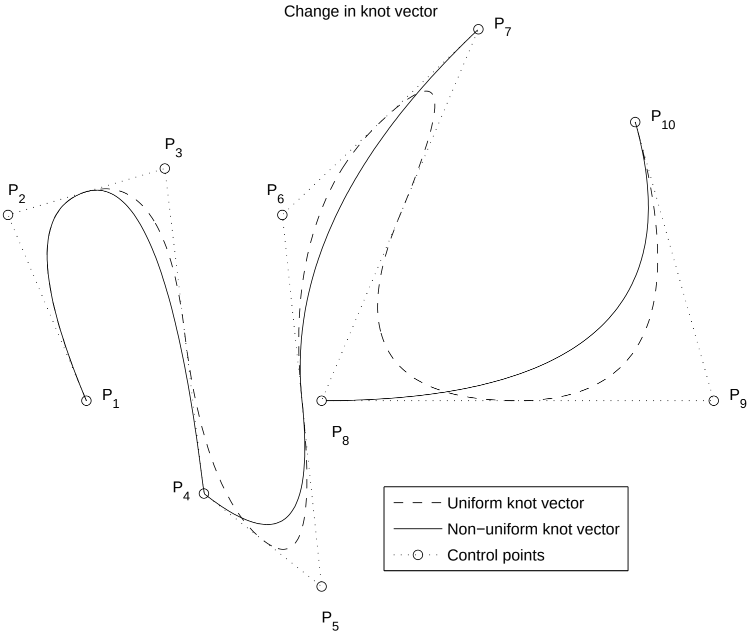

Figure 5. Influence of the change in the knot vector to the shape of curve. In the curve with a uniform knot vector the parameters are same as in Example 1. Only change between the curves is that the second one has a non-uniform knot vector. The non-uniform knot vector of the curve in this figure is: $U = \{0,0,0,\frac{1}{8},\frac{2}{8},\frac{2}{8},\frac{4}{8},\frac{6}{8},\frac{6}{8},\frac{6}{8},1,1,1\}$.

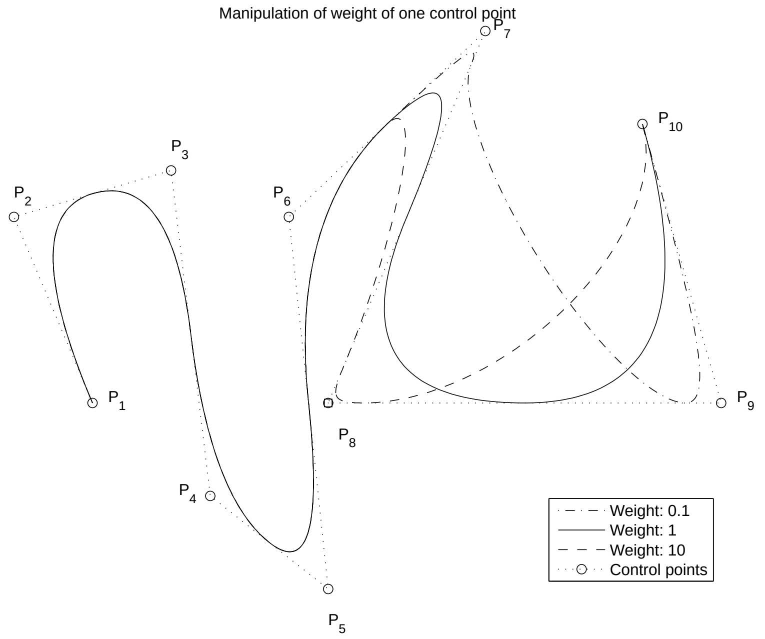

Figure 6. The manipulation of the weight affects to the curve. All other parameters are the same as in Example 1 except the weight of the control point P_8 is changed.

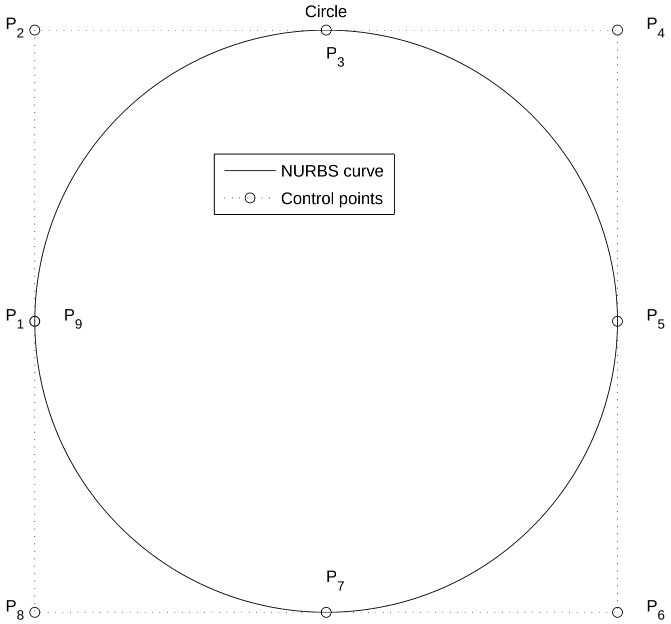

Figure 7. Perfect circle: created using NURBS.

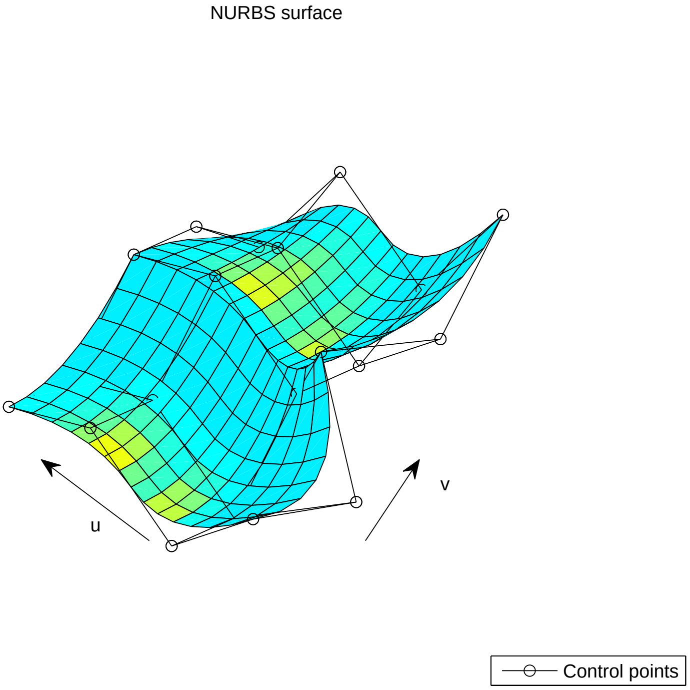

Figure 8. Arbitrary NURBS surface

4 PERFORMANCE TESTS

Our NURBS toolbox is compared to two other toolboxes (toolbox A [20] and toolbox B [21]). In the performance tests we were intrested in the computational times used by the various toolboxes. 24 different test cases were run with the three toolboxes. The test cases are presented in Tables 1, 2, 3, 4, 5 and 6. All computational times are measured in seconds. The control points of the test cases where uniformly distributed random numbers between zero and one. The weights were a uniformly distributed random numbers between 0.5 and 1.5. The numbers of the parametric points and the control points are presented in Tables 1, 2, 3, 4, 5 and 6. Each table includes four test cases. Each test case includes three numbers. These numbers forms always a row. First number on each row is the time of toolbox A. Second number is time of the NURBS toolbox and third number is a time of the toolbox B.

Tables 1 and 2 shows times for evaluation of curves. Tables 3 and 4 shows times for evaluation of surfaces (net of parametric points) and Tables 5 and 6 shows times for evaluation of surfaces (scattered parametric points). It can be seen from tables that when the number of the parametric points increased the time used increased also. If order increased evaluation time increased too. Toolbox B was slowest toolbox in all test cases. The computation times of the NURBS toolbox and the toolbox A were very similar when curves were evaluated. When surfaces (net of parametric points) were evaluated the toolbox A was bit faster than the NURBS toolbox. The NURBS toolbox was clearly the fastest toolbox when the scattered parametric point surfaces were evaluated. It seems that the NURBS toolbox is efficient when comparing it to toolboxes A and B.

Table 1: Curve evaluation comparison between the NURBS toolbox, the toolbox A and the toolbox B when order is 3. Computational times are in seconds.

Toolbox		**A**	**NURBS**	**B**	**A**	**NURBS**	**B**
Number of parametric points		**1000000**			**4000000**		
Number of control points	**100**	0.93	0.82	17.51	3.84	3.30	70.63
	10000	1.14	0.83	18.18	4.71	3.34	72.51

Table 2: Curve evaluation comparison between the NURBS toolbox, the toolbox A and the toolbox B when order is 8. Computational times are in seconds.

Toolbox		**A**	**NURBS**	**B**	**A**	**NURBS**	**B**
Number of parametric points		**1000000**			**4000000**		
Number of control points	**100**	1.89	2.05	25.33	7.63	8.21	101.83
	10000	2.12	2.09	25.89	8.52	8.34	102.60

Table 3: Surface evaluation (net of parametric points) comparison between the NURBS toolbox, the toolbox A and the toolbox B when order is 3. Computational times are in seconds.

Toolbox		**A**	**NURBS**	**B**	**A**	**NURBS**	**B**
Number of parametric points		**1000000**			**4000000**		
Number of control points	**100**	0.47	0.37	2.09	1.86	1.54	8.09
	10000	0.51	0.76	2.28	1.91	3.29	8.28

Table 4: Surface evaluation (net of parametric points) comparison between the NURBS toolbox, the toolbox A and the toolbox B when order is 8. Computational times are in seconds.

Toolbox		**A**	**NURBS**	**B**	**A**	**NURBS**	**B**
Number of parametric points		**1000000**			**4000000**		
Number of control points	**100**	0.66	0.62	2.57	2.65	2.53	10.08
	10000	0.77	1.14	2.85	2.73	4.38	10.57

Table 5: Surface evaluation (scattered parametric points) comparison between the NURBS toolbox, the toolbox A and the toolbox B when order is 3. Computational times are in seconds.

Toolbox		**A**	**NURBS**	**B**	**A**	**NURBS**	**B**
Number of parametric points		**100000**			**400000**		
Number of control points	**100**	3.22	0.18	9.14	5.16	0.30	26.87
	10000	5.24	0.30	27.07	21.18	1.08	107.05

Table 6: Table 6: Surface evaluation (scattered parametric points) comparison between the NURBS toolbox, the toolbox A and the toolbox B when order is 8. Computational times are in seconds.

Toolbox		**A**	**NURBS**	**B**	**A**	**NURBS**	**B**
Number of parametric points		**100000**			**400000**		
Number of control points	**100**	3.59	0.74	11.00	14.10	2.96	43.48
	1000	7.13	0.82	33.42	28.52	3.37	132.20

5 DISCUSSION

The purpose of this work was to implement a NURBS toolbox to Matlab environment. This ultimate goal was reached. The current version of the NURBS toolbox is 2.3. The NURBS toolbox includes three functions for the user. By using these functions the user can compute NURBS curves and surfaces. The data structures used in Matlab are easy to understand. All parameters of NURBS are set inside one structure array. This eases the handling of the parameters of the NURBS curve or surface. The NURBS toolbox is implemented so that it can be expanded easily. This is a good property for a user who wants to add something to the NURBS toolbox in future. The NURBS toolbox can be used for evaluation of NURBS curves or surfaces. NURBS toolbox can be used as a part of a larger computing system. This is possible because NURBS toolbox provides functions which can be used wherever NURBS are needed.

5.1 Future work

NURBS toolbox to Matlab environment was implemented. The Current version number of the NURBS toolbox is 2.3. The NURBS Toolbox allows a user to manipulate the parameters of NURBS curve or NURBS surface. NURBS Toolbox is now ready for use, modifications, and extensions. We think that algorithms of NURBS toolbox could be better. In future at least some of the algorithms could be improved to get more efficient toolbox for computing NURBS. The NURBS Toolbox works now in Matlab environment but all important algorithms have written using C-language. These source codes are compatible with mex-files. It would not be a big task to make the NURBS toolbox work also without Matlab. If the NURBS toolbox would work without Matlab it would improve the computational efficiency of the NURBS toolbox.

In addition of improving the algorithms of the NURBS toolbox it could be extended. First extension could be the knot insertion algorithm. Using the knot insertion algorithm it is possible to add more control points to the existing NURBS curve or NURBS surface. With the good knot insertion algorithm adding could be done so that the shape of the NURBS curve or the NURBS surface would not change. Similarly, the knot removal algorithm would be useful. With this it would be possible to remove control points from the already existing NURBS curve or NURBS surface. Removing should be done so that shape of curve or surface changes only near the removed control point.

It is possible to compute the derivatives and integrals of NURBS curves or NURBS surfaces. The possibility to evaluate derivatives or integrals could be added to the NURBS toolbox. It would be nice to be able to approximate set of points with NURBS. A tool of this kind could be added to the NURBS toolbox. This kind of approximation could be done for both curves and surfaces. The NURBS Toolbox is now providing only evaluation of NURBS curve or NURBS surface. Some kind of visualiation tool could be done to make the understanding of NURBS shape easier. A visual editing tool for NURBS control points could be also done.

6 SUMMARY

The purpose of this work was to implement NURBS toolbox to Matlab environment. Requirements for this NURBS toolbox was that it should efficient, easy to use, and extensible. In the implementation we decided to use already existing algorithms with possible improvements. NURBS can presents both curves and surfaces. NURBS are parametric curves or surfaces. By manipulating the parameters of NURBS it can get very different shapes. The parameters of NURBS are control points, weights, a knot vector and an order. NURBS has been firstly used in CAD applications. In addition to CAD applications NURBS are used for example in the field of medical science. Matlab is a programmable computation environment.

The user interface of this NURBS toolbox consists of three functions. Two of these functions are for the evaluation of a NURBS surface, one is for the evaluation of a NURBS curve. The first of the surface evaluation functions is for the evaluation of a net of parametric points. The second surface evaluation function is meant for evaluating scattered surface points which are not related to the each other. Our NURBS toolbox consists of m-files, mex-file and c-files. A user is using the toolbox via m-files, they m-files are passing the parameters to mex-files where the actual computation happens. Mex-files are compiled from c-files so that the actual algorithms of the NURBS toolbox were written using C-language. This makes it more efficient compared to pure Matlab applications.

Examples in Chapter 3 help understanding NURBS and also the use of the NURBS toolbox. When first learnt how to NURBS curves are manipulated then is easy to learn how to manipulate NURBS surfaces. At least manipulation of weight of control point affect intuitively to the shape of the surface. Whole NURBS or parts of that can be translated or stretched by manipulating the control points and the weights of NURBS.

NURBS toolbox was compared to other two NURBS toolboxes. The Measured parameter was the running time. We used 24 different test cases. First of the other two toolboxes was clearly the most unefficient of all three toolboxes. The efficiency comparison between NURBS toolbox and the second of the other toolboxes was tighter and both toolboxes were good at some tasks according to the comparisons. The NURBS Toolbox has piled up so that it can be further developed.

REFERENCES

[1] Les Piegl and Wayne Tiller. *The NURBS Book*. Springer, 1995.

[2] Les Piegl. On NURBS: A survey. *Computer Graphics and Applications, IEEE*, 11:55–71, 1991.

[3] W.P. Segars, D.S. Lalush, and B.M.W. Tsuil. A realistic spline-based dynamic heart phantom. *Nuclear Science, IEEE Transactions on*, 46:503–506, 1999.

[4] William C. Regli Jie Li and Wei Sun. Mathematical representation of the vascular structure and applications. In *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 373 – 378, Beijing, China, June 2007.

[5] M. G. Cox. The numerical evaluation of b-splines. *J. Inst. Maths Applics*, 10:134–149, 1972.

[6] Carl De Boor. *A Practical Guide to Splines*. Springer-Verlag, 1978.

[7] W. Tiller. Rational b-splines for curve and surface representation. *IEEE Comput. Graph. and Appl.*, 3(49):61–69, 1983.

[8] G. E. Farin. *Curves and Surfaces for Computer Aided Geometric Design - A Practical Guide*. Boston: Academic Press, 1993.

[9] Gerald Farin, editor. *NURBS for Curve and Surface Design*. Philadelphia (PA) : Society for Industrial and Applied Mathematics, 1991.

[10] John Lowther John Fisher and Ching-Kuang Shene. If you know B-splines well, you also know NURBS! In *Proceedings of the 35th SIGCSE technical symposium on Computer science education*, pages 343 – 347, Norfolk, Virginia, USA, March 2004.

[11] I. D. Faux and M. J. Pratt. *Computational Geometry for Design and Manufacture*. Chichester, UK: Ellis Horwood Ltd., 1981.

[12] J. Perez, J.A. Saiz, O.M. Conde, R.P. Torre, and M.F. Catedra. Analysis of antennas on board arbitrary structures modeled by nurbs surfaces. *Antennas and Propagation, IEEE Transactions*, 45:1045–1053, 1997.

[13] P. Lavoie, D. Ionescu, and E. Petriu. Constructing 3D virtual reality objects from 2D images of real objects using NURBS. In *Virtual Environments, Human-Computer Interfaces and Measurement Systems*, pages 117 – 124, Ostuni, Italy, 25-27 June, 2007.

[14] In Kyu Park, Il Dong Yun, and Sang Uk Lee. Constructing NURBS surface model from scattered and unorganized range data. In *3-D Digital Imaging and Modeling, 1999. Proceedings*, pages 312 – 320, Ottawa, Canada , 4-8 Oct. 1999.

[15] Song Han and Gerard Medioni. Triangular NURBS surface modelling of scattered data. In *Visualization '96. Proceedings*, pages 295–302, San Francisco, 27 Oct.-1 Nov., 1996.

[16] C. Bajaj, H. Y. Lee, R. Merkert, and V. Pascucci. NURBS based B-rep models for macromolecules and their properties. In *Proceedings of the Fourth ACM Symposium on Solid Modeling and Applications*, pages 217–228, Atlanta, Georgia, United States, May 14-16, 1997.

[17] Arto Kaarna. NURBS representation for sets of spectral signatures. In *Proceedings of IEEE Geoscience and Remote Sensing Symposium*, pages 136–139, Seoul, Korea, 25-29 July 2005.

[18] Segars W.P., Lalush D.S., and Tsuil B.M.W. The next generation MCAT: A realistic spline-based torso phantom. *Engineering in Medicine and Biology*, 2:978, 1999.

[19] Jia Lu Xianlian Zhou. NURBS-based Galerkin method and application to skeletal muscle modeling. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*, pages 71 – 78, Cambridge, Massachusetts, June 2005.

[20] NURBS R2006b. From: http://www.mathworks.com/matlabcentral/fileexchange/12558, Retrieved: November 20th 2008.

[21] Converted NURBS toolbox. From: http://www.mathworks.com/matlabcentral/ fileexchange/14247, Retrieved: November 20th 2008.

NURBS curve evaluation example

```
nurbs.p=4;
nurbs.U=[0 0 0 0 .3 .3 .3 .3 .7 .7 1 1 1 1];
nurbs.Pw=[0  2  2  0  0  5  5  3  3  5;  %(x1)
          0  0  3  3  6  6  3  3  0  0;  %(y1)
          1  1  1  1  0  0  1  1  1  1;  %(z1)
          1  1  1  1  2  1  1  1  1  1;  %(weights1)

          1  2  3  4  5  5  4  3  2  1;  %(x2)
         -1  0  1  0  0  4  7  8 -2  2;  %(y2)
          0  0  0  0  0  0  0  0  0  0;  %(z2)
          1  1  1  1  2  1  1  1  1  1]; %(weights2)

C = nurbsCurve(nurbs, 3, [0 0.25 0.5 0.75 1])

Result of the example:

C =
         0    0.8333    3.7619    3.5053    5.0000
         0    2.7778    4.6667    2.7778   -0.0000
    1.0000    1.0000    0.4444    1.0000    1.0000
    1.0000    3.5000    4.4921    3.1693    1.0000
   -1.0000    0.3426    4.5079    7.0298    2.0000
         0         0         0         0         0
```

NURBS surface evaluation example

```
nurbs.p=3;
nurbs.U=[0 0 0 1 1 1];
nurbs.q=4;
nurbs.V=[0 0 0 0 1 1 1 1];

nurbs.Pw(:,:,1)=[0 1 2;        nurbs.Pw(:,:,2)=[0 1 2;
                 0 0 0;                         1 1 1;
                 0 0 0;                         0 0 0;
                 1 1 1];                        1 1 1];

nurbs.Pw(:,:,3)=[0 1 2;        nurbs.Pw(:,:,4)=[0 1 2;
                 2 2 2;                         3 3 3;
                 0 3 0;                         0 0 0;
                 1 1 1];                        1 1 1];

S=nurbsSurfaceCheck(nurbs, {[0 .5 1] [0 .5 1]})

Result of the example:

S(:,:,1) =                   S(:,:,2) =
     0     1     2                    0    1.0000    2.0000
     0     0     0               1.5000    1.5000    1.5000
     0     0     0                    0    0.5625         0

S(:,:,3) =

     0     1     2
     3     3     3
     0     0     0
```