A novel approach to designing optical devices based on NURBS surfaces

Edgar Teufel

German Aerospace Center, Institute of Technical Thermodynamics, Pfaffenwaldring 38-40, D-70569 Stuttgart

Abstract

Optical system design has a significant impact on solar power plant efficiency. It can be advantageous to be able to very precisely adapt mirror shapes. In this article it is shown how NURBS surfaces can be used as mirrors by lining out the equations and algorithms needed to evaluate their optical performance by ray tracing. A parameterization scheme is given that reduces the degrees of freedom to a level suitable for automatic processing. A secondary reflector based on a NURBS surface is designed and adapted to a test application. By demonstration it is proved that this is feasible on modern PCs. The new concentrator's performance is calculated and compared to a CPC. It is found that the fine-tuned NURBS reflector outperforms the CPC.

Keywords: concentrating solar power, secondary concentrators, CPC, ray tracing, optical performance

1. Introduction

Designing optical systems for solar power tower plants and other applications is a challenging task. One wants optical devices to collect as much solar irradiation as possible and focus it to the maximum possible extent without absorbing significant fractions of it. Moreover, such concentrators shall be cheap to manufacture and last for many years in harsh environments. But, given that such optical systems influence annual yield of solar power plants directly and that their cost does not depend on the design of the shape in the first place, almost every effort of finding even small improvements in shape efficiency is worth it.

Historically, analytical rather than numerical approaches have been followed to tackle the problem of finding good concentrator shapes. Good success has

☆This work has been partly funded through a grant emitted by the European Commission under its Seventh Framework Programme, project acronym SFERA, grant agreement no.: 228296.

Email address: edgar.teufel@dlr.de (Edgar Teufel)

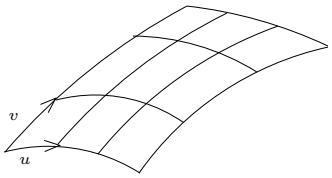

Figure 1: Coordinates (u, v) of the NURBS surface

been achieved using parabolics, cones or even combined shapes like the the compound parabolic concentrator (CPC) suggested by Winston (1974). Concentrators with non-regular shapes have also been presented (see Timinger et al. (2000) and Kribus et al. (2000)). However, even the two latter articles jump relatively short in terms of non-regularness as they "only" add distortion to the well-known CPC and a concentrator shape with rectangular or polygonal cross section to the collective of investigated shapes. The reader may be felt left back with a vague suspect that some improvements can still be achieved when substantially more degrees of freedom for the shape are available. And the conclusion of Timinger et al. (2000) that the concentrators with rectangular cross section totally unexpectedly showed better performance than the smoothly shaped concentrators only adds to this impression.

In this contribution, NURBS (non-uniform rational Béziers spline) surfaces are used as shapes for secondary concentrators. NURBS surfaces can have virtually any shape. They are state of the art in free-form surface modelling and gained wide acceptance in computer graphics for their intuitive and flexible shape editing possibilities. One negative aspect of NURBS is their relative mathematical complexity. For example they are defined only recursively which makes it difficult to carry out algebraic calculations. Moreover, numeric calculations with NURBS are computationally quite expensive. Both aspects have to be addressed when one wants to use NURBS surfaces as optical devices.

2. NURBS surfaces

NURBS can be seen as an extension of the "spline" concept for smooth two-dimensional curves to three dimensions. Piegl and Tiller (1996) give a good overview and introduction to the topic. Much like flat, two-dimensional curves can be defined in a parametric way depending on just one running variable t, NURBS surfaces are defined over a two-dimensional domain with parameters u and v, whose values are in the interval $[0, 1]$. This reflects that topologically NURBS surfaces always have a planar character (see fig. 1).

In its most generic form, a point $\overrightarrow{S}$ on a NURBS surface with (surface) coordinates (u, v) is by definition given as interpolated and weighted average of the n by m control points $\overrightarrow{P}$:

$$\overrightarrow{S}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}\overrightarrow{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}}. \quad (1)$$

A weight w is associated with each control point. The basis functions N with degrees p and q in u- and v-direction respectively do the interpolation of the surface between the control points. They are defined as follows:

$$N_{i,0}(u) = \begin{cases} 1 & \forall \quad u_i < u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u), \quad (3)$$

with $\mathbf{U}$ being the so called knot vector. The knot vector contains $n + p + 1$ elements u_i in nondecreasing order and determines which control points affect the surface shape in which area. Usually, there is a knot vector $\mathbf{U}$ in u-direction and $\mathbf{V}$ in v-direction.

Note how the basis function is defined recursively. A quick algorithm to calculate N is given by Piegl and Tiller (1996) (see Algorithm A2.4, pp. 74-75).

3. Ray-tracing with NURBS surfaces

When it comes to assessing the performance of optical systems, tracing of rays through the system is a very successful method. More specifically, this means that intersection points of arbitrary rays with the surface have to be calculated along with the surface normals at these locations. Given the recursive definition of the NURBS surface, this obviously is not a trivial task.

3.1. Surface normal

Calculating the direction of a reflected ray requires to know the direction of the surface normal $\overrightarrow{N}$ at the intersection point (u, v) of the surface with the incoming ray. This, in turn, needs the partial derivatives of $\overrightarrow{S}(u, v)$ with respect to u and v to be known. Applying the quotient rule to eq. (1) yields:

$$\begin{aligned}
\overrightarrow{S}_u(u,v) &:= \frac{\partial \overrightarrow{S}(u,v)}{\partial u} = \frac{\overrightarrow{c}_1 - \overrightarrow{c}_2}{\left(\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\right)^2}, \\
\overrightarrow{c}_1 &= \sum_{i=0}^{n}\sum_{j=0}^{m} N'_{i,p}(u)N_{j,q}(v)w_{i,j}\overrightarrow{P}_{i,j} \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v), \quad (4) \\
\overrightarrow{c}_2 &= \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}\overrightarrow{P}_{i,j} \sum_{i=0}^{n}\sum_{j=0}^{m} N'_{i,p}(u)N_{j,q}(v).
\end{aligned}$$

A similar expression symmetrical to this one can be given for $\overrightarrow{S}_v$. The derivate of $N_{i,p}(u)$ with respect to u, abbreviated with $N'_{i,p}(u)$ in eq. (4), is given by Piegl and Tiller (1996) (see Algorithm A2.5, pp. 76-78) along with a quick algorithm to compute it.

Now, with both $\overrightarrow{S_u}$ and $\overrightarrow{S_v}$ at hand, computing $\overrightarrow{N}$ is straightforward:

$$\overrightarrow{N} = \overrightarrow{S_u} \times \overrightarrow{S_v}. \quad (5)$$

3.2. Ray-surface-intersection

A ray $\overrightarrow{r}$ originating from $\overrightarrow{O}$ and emerging along direction $\overrightarrow{D}$ with parameter t may be defined as follows:

$$\overrightarrow{r}(t) = \overrightarrow{O} + t\overrightarrow{D}. \quad (6)$$

As for any intersection point the identity $\overrightarrow{r}(t) = \overrightarrow{S}(u,v)$ must hold, one could just insert eq. (6) into eq. (1) and iteratively search triples of (t,u,v). That would solve the problem. However, there is a better approach suggested by Martin et al. (2000). The idea is to express the ray as intersection of two planes, that are constructed using the intersection point of the ray with the NURBS surface. This way, only two parameters have to be determined iteratively.

The Hesse normal form of a plane E is given by

$$\overrightarrow{p} \cdot \overrightarrow{N_0} - d = 0, \quad (7)$$

where $\overrightarrow{p}$ is a location vector to a point $P \in E$, $\overrightarrow{N_0}$ is the normal vector of E of unit length and d is the distance of E from the origin. Applying this to our problem yields two planes E_1 and E_2:

$$\begin{aligned}
E_1 : \overrightarrow{p} \cdot \overrightarrow{N_1} - d_1 &= 0 \quad (8) \\
E_2 : \overrightarrow{p} \cdot \overrightarrow{N_2} - d_2 &= 0 \quad (9)
\end{aligned}$$

Martin et al. (2000) suggest for the normal vector $\overrightarrow{N_1}$ to choose

$$\overrightarrow{N_1} = \begin{cases} (D_y, -D_x, 0) & \text{if } |D_x| > |D_y| \text{ and } |D_x| > |D_z| \\ (0, D_z, -D_y) & \text{otherwise} \end{cases} \tag{10}$$

to ensure that $\overrightarrow{N_1}$ is always perpendicular to the ray direction $\overrightarrow{D}$. $\overrightarrow{N_2}$ then just becomes

$$\overrightarrow{N_2} = \overrightarrow{N_1} \times \overrightarrow{D}. \tag{11}$$

As both E_1 and E_2 must contain the origin $\overrightarrow{O}$ of the ray by definition, the distances d_1 and d_2 can be calculated as

$$d_1 = \overrightarrow{N_1} \cdot \overrightarrow{O} \tag{12}$$

$$d_2 = \overrightarrow{N_2} \cdot \overrightarrow{O}. \tag{13}$$

Now that the two planes E_1 and E_2 that contain the ray are known and bearing in mind that the intersection point to be calculated must also satisfy these plane equations, the problem of finding the (u, v)-coordinates of it reduces to finding the roots of

$$\mathbf{F}(u, v) = \begin{pmatrix} \overrightarrow{N_1} \cdot \overrightarrow{S}(u, v) - d_1 \\ \overrightarrow{N_2} \cdot \overrightarrow{S}(u, v) - d_2 \end{pmatrix}. \tag{14}$$

There are various numerical methods that can theoretically be applied to the problem. However, following the advice given by Press et al. (2002) in their excellent reference, only the Newton method is likely to reliably give a good solution - with the downside of this choice being that the Newton method needs a good initial estimate. We will deal with that later.

Applying the Newton method to eq. (14) immediately yields the iteration law for the intersection point coordinates (u, v):

$$\begin{pmatrix} u_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} u_n \\ v_n \end{pmatrix} - \mathbf{J}^{-1}(u_n, v_n) \cdot \mathbf{F}(u, v), \tag{15}$$

with $\mathbf{J}$ being the Jacobian matrix defined as:

$$\mathbf{J} = \begin{bmatrix} \overrightarrow{N_1} \cdot \frac{\partial S}{\partial u} & \overrightarrow{N_1} \cdot \frac{\partial S}{\partial v} \\ \overrightarrow{N_2} \cdot \frac{\partial S}{\partial u} & \overrightarrow{N_2} \cdot \frac{\partial S}{\partial v} \end{bmatrix} = \begin{bmatrix} \overrightarrow{N_1} \cdot S_u & \overrightarrow{N_1} \cdot S_v \\ \overrightarrow{N_2} \cdot S_u & \overrightarrow{N_2} \cdot S_v \end{bmatrix}. \tag{16}$$

3.3. Reflection

Computing direction and origin of a ray reflected on the reflective side of the NURBS surface needs all the material outlined in sections 3.1 and 3.2 plus some additional conditions and criteria developed in the following.

Before the Newton method can be used to iteratively calculate the intersection point coordinates of some incoming ray with the NURBS surface, an initial guess is required. This initial guess must be good enough for the Newton

method to converge to the correct, i.e. nearest to the ray's origin, intersection point - if any. Note that there might be zero, one or multiple intersection points. There are many ideas how this problem can be tackled (see Abert et al. (2006) and Nishita et al. (1990)). As in the context of secondary reflectors for solar power tower applications the shapes topologically are significantly less complex compared to the ones commonly discussed in literature, a simpler approach is followed in this contribution. For the purpose of calculating a first guess for the intersection point coordinates, the NURBS surface can be approximated with triangles. Note that because of the definition of (u, v) (see fig. 1) it is relatively straighforward to obtain the edge point coordinates of triangles such that the discrete grid covers the smooth surface adequately for example by distributing discrete (u_i, v_i) equally in u- and v-direction of the surface. Of course, this discretization has to be done only once for each shape. Then, one can intersect the incoming ray with each of these triangles (a process which virtually cries for parallelization by the way...) and use the edge-point-averaged (u, v) coordinates of the hit triangle closest to the ray's origin in downstream direction as initial guess.

With this initial guess, the Newton method will converge quickly towards the exact intersection point. As a stop criterion, one can use the normed distance of the current point from the two planes representing the incoming ray:

$$\|\mathbf{F}(u_n, v_n)\| < \epsilon \tag{17}$$

During the whole iteration process, it has also to be made sure that the (u_n, v_n) do not fall outside of the definition range and that the Jacobian matrix does not get singular. In the latter case, one can for example step randomly 0..10% of the distance towards the initial guess. In rare cases it might also occur a situation where the intial guess calculation reports a hit where there in fact is none. In this case, the Newton method would search forever, which should be avoided, too.

With the coordinates (u_S, v_S) of the intersection point, the direction of the surface normal $\overrightarrow{N}$ can be calculated following the algorithm outlined in section 3.1. If the incoming ray's direction is $\overrightarrow{r}_{in}$, the reflected ray's direction $\overrightarrow{r}_{out}$ becomes:

$$\overrightarrow{r}_{out} = \overrightarrow{r}_{in} - 2(\overrightarrow{r}_{in} \cdot \overrightarrow{N})\overrightarrow{N}. \tag{18}$$

The only thing left is to check whether or not the incoming ray hits the surface from the reflective side. One possible solution to this is to evaluate the angle between $\overrightarrow{N}$ and $\overrightarrow{r}_{in}$:

$$\begin{aligned} \alpha &:= \angle(\overrightarrow{N}, \overrightarrow{r}_{in}) \\ \alpha > \frac{\pi}{2} &\rightarrow \text{reflective side hit} \end{aligned} \tag{19}$$

Eventually, moving the reflected rays origin some small distance downwards

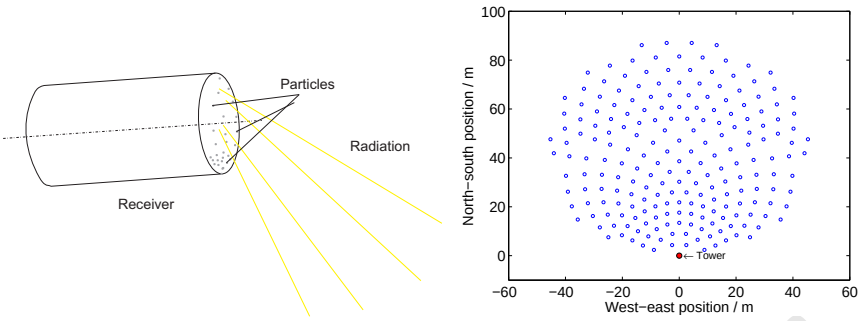

Figure 2: Rotating tube receiver and heliostat field

the ray's emerging direction helps a lot in reducing erroneous self-intersections due to rounding errors.

Now that the tools for evaluating NURBS surfaces as reflector are lined out, the problem of designing a secondary reflector with them that is especially well suited for discrete situations can be tackled. We will define a test application setup first.

4. Test application setup

At the heart of the test application is a new receiver design (see fig. 2, left part). Rather than heating up some fluid, this receiver works with solid particles. At its entrance, small particles falling from the top of the rotating tube of the receiver down to the bottom form some sort of curtain which will be heated up by the incoming radiation. The hot particles will then be transported away from the entrance aperture towards a heat exchanger. Transportation of the particles takes place because the whole receiver tube rotates around its symmetry axis.

As far as optical performance is concerned, this special receiver design is quite a challenge. In order to prevent the particles from falling out of the rotating tube, the receiver has to be directed slightly towards the sky (in this case by 4°) instead of towards the incoming radiation. Therefore, its entrance aperture has to be much larger to compensate for the "projection losses". However, this leads to significantly elevated radiation and convection losses. Thus, one is tempted to add some optical device that would "bent" the incoming radiation by some 40°- and ideally also concentrate it if possible.

The right part of fig. 2 depicts the heliostat field of the test application. It has been calculated along with the altitude of the receiver aperture center using some of the plant properties listed in table 1 and modified slightly by hand (some heliostats located south of the tower receiver have been removed).

The "figure of merit" or cost function value to quantify the performance of the application is an annual yield estimation that is quick to compute. For each full hour between sunrise and solar noon and for solar noon itself the current power absorbed by the receiver is calculated. The absorbed power P_{abs}

Property	**Value**
Design point	
Date and time	June 21st at solar noon
Radiative power in receiver aperture	2 MW
Geological position	
Latitude	48.2°
Elevation	550 m
Heliostats	
Facet size (width x height)	2.56 m x 3.22 m
Facets	1x1
Canting	not applicable
Focus	slant range
Reflectivity	87%
Tracking type	Azimuth-Elevation
Mirror slope error (including sunshape error)	3.67 mrad
Elevation axis altitude (pedestal height)	2.0 m
Receiver	
Type	flat
Diameter	2.0 m
Orientation	tilted downwards by 4°
Altitude entrance aperture center	27.5 m

Table 1: Plant properties of the test application

is assumed to be the collected radiative power P_{rad} minus five percent to model reflection losses minus re-radiation and convection losses. The latter two are summarized and modelled to be proportional to the receiver area A_{rec}:

$$P_{abs} = 0.95 P_{rad} - 187.0 \frac{kW}{m^2} A_{rec}, \tag{20}$$

where the value $187.0 kW/m^2$ is the sum of $100.0 kW/m^2$ which is an rule of thumb estimation of the specific convection losses and $87.0 kW/m^2$ which is what a black body radiator with surface temperature 840°C would emit according to the Stefan-Boltzman law. However, if these losses exceed 95% of P_{rad}, P_{abs} is set to zero by definition as in that case one would not operate the receiver.

This power estimation calculation is done for three days: March 21st, June 21st and December 21st. From these figures, an "annual average current power" is calculated by forming a weighted average, thereby making sure that the non solar noon values count twice as much as the solar noon value to reflect day symmetry and that the March value counts twice as much as the December and June values to reflect annual symmetry. Multiplying the resulting value with the daylight hours per year finally delivers the annual yield estimation.

5. Parameterization and setup of the NURBS surface

Reading the test application setup specified in section 4, one intuitively thinks that some kind of secondary concentrator shaped like a "nozzle" could improve the optical performance here. So we are talking about a closed shape with a topology similar to that of a bent tube. Thus, we are after a NURBS surface with circular cross-section which very much suggests to distribute nine control points in circumferal direction. Without compromising flexibility too much, one can decide to also use nine control point lines in lateral direction and to set the base function degree in both directions to two. So we have (see eq. 1):

$$\begin{aligned} n = m &:= 9 \\ p = q &:= 2 \end{aligned}$$

This choice mutually also leads to a knot vector of length $n + p + 1 = 12$ that again can be used both in u- and in v-direction. It can take these values:

$$\mathbf{U} = \mathbf{V} = \{0.0, 0.0, 0.0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1.0, 1.0, 1.0\}.$$

The only things left to do now are the definition of the control points $\overrightarrow{P}_{i,j}$ and the weights $w_{i,j}$. One could think of using these values as degrees of freedom directly and let them be adopted to maximize the test application cost function automatically. However, this would overcharge available search algorithms by far. The huge number of degrees of freedom along with the computationally quite expensive cost function evaluation would lead to unrealistic calculation times. Therefore, the setting up of control point coordinates and weight values must be performed by some model with far less parameters.

The idea of the parameterization is to arrange the NURBS control points along some path, the "control point path", modelled as parametric Bézier curve, and use the (x, y) coordinates of the control points (P_1, P_2, P_3) of this path as parameters:

$$\begin{pmatrix} C_x \\ C_y \end{pmatrix}(t) = \left[3(1-t)^2 t \begin{pmatrix} P_{1,x} \\ P_{1,y} \end{pmatrix} + 3(1-t)t^2 \begin{pmatrix} P_{2,x} \\ P_{2,y} \end{pmatrix} + t^3 \begin{pmatrix} P_{3,x} \\ P_{3,y} \end{pmatrix} \right] l.$$

When t runs from zero to one, the cross section center point C of the final shape runs from $(0|0)$ to lP_3, with l being used as scale factor representing the lateral length of the shape. Thus, the complete path of the NURBS shape's cross section center point can be described with only seven parameters.

The same technique can be used to parameterize the radial extent of the shape. The "radius path" shall specify the radius of each cross section along the control point path. As the exact shape of the reflector surface at the exit aperture end is crucial for optical performance, it seems advantageous to throw in some more degrees of freedom here. Therefore, this path is divided into two

"arms", with each of them being described by a quadratic Bézier curve. The left arm shall run from $(0|0)$ towards R_1 and end in R_M. The right arm shall start in R_M towards R_2 and end in R_3. R_M shall thereby lie on the line between R_1 and R_2 at a distance from R_1 specified by a "proximity factor" f_{prox}. So we have:

$$\begin{aligned} \begin{pmatrix} R_{M,x} \\ R_{M,y} \end{pmatrix} &= (1 - f_{prox}) \begin{pmatrix} R_{1,x} \\ R_{1,y} \end{pmatrix} + f_{prox} \begin{pmatrix} R_{2,x} \\ R_{2,y} \end{pmatrix} \\ r &= \begin{cases} 2t(1-t)R_{1,y} + t^2 R_{M,y} \\ (1-t)^2 R_{M,y} + 2t(1-t)R_{2,y} + t^2 R_{3,y}. \end{cases} \end{aligned}$$

The dimensionless preliminary radius r is converted into an interim value r_i by scaling it with a factor f_s:

$$r_i = (1 + r) f_s.$$

Up to this point, the cross section of the concentrator shape is circular. Making it irregular for example by allowing ovalized forms by shifting the edge control points a bit could possibly improve optical performance. This can be accomplished by defining the distance r_{CP} of each control point from the control point path as follows:

$$r_{CP} = \begin{cases} r_i r_{ex} f_o & -j \text{ even and } j \in \{2, 6\} \\ r_i r_{ex} / f_o & -j \text{ even and } j \notin \{2, 6\} \\ \sqrt{(r_i r_{ex} f_o)^2 + (r_i r_{ex})^2 / f_o^2} & -\text{otherwise} \end{cases},$$

with r_{ex} being the nominal radius of the exit aperture and f_o being an ovality factor. This also binds the index j to the circumferal, i.e. v-direction. Both indices (i, j) are elements of the set $\{0, 1, 2, \ldots, 8\}$.

Thus, the calculation of the NURBS control points $\overrightarrow{P}_{i,j}$ works as follows. The index i is looped in an outer loop. For each i a t_1 is calculated ($t_1 = i/8$). With this t_1 the control point path center point C is calculated (in the (x, y)-domain). For the obtained C_x another t_2 is calculated that corresponds with a value in the correct arm of the radius path. Evaluating this t_2 gives an interim radius r_i. Looping the index j in an inner loop, evaluation of r_i leads to r_{CP} and fills the control point ring. Then this ring has to be converted to the (x, y, z)-domain by rotating it around its diameter about an angle corresponding to the slope of the control point path at the current $i/8$ and moving it to C.

Now for the weights $w_{i,j}$. Their values shall depend only on j. With an additional factor r_{rect} representing "rectangularness" they can be defined as follows:

$$w_{i,j} = \begin{cases} 1.0 & -j \text{ even} \\ f_{rect}\sqrt{2}/2 & -\text{otherwise} \end{cases}.$$

Name	Value range	Comment
l	2.0..5.0	length scale factor (unit: meter)
f_{dist}	0.01..1.0	$P_{1,x} = f_{dist}\cos(\delta+\gamma)$ $P_{1,y} = f_{dist}\sin(\delta+\gamma)$ $\delta = 4°$- tilt angle of receiver aperture
$P_{2,x}$	0.0..0.8	
$P_{2,y}$	-0.1..0.1	
$P_{3,y}$	-0.3..0.3	$P_{3,x} := 1$
$R_{1,x}$	0.1..0.7	
$R_{1,y}$	0.1..1.0	
$R_{2,x}$	0.2..0.9	
$R_{2,y}$	0.1..1.0	
$R_{3,y}$	0.1..3.0	$R_{3,x} := 1$
f_{prox}	0.01..0.99	
$f_{o,entr}$	0.5..2.0	$f_o = t f_{o,entr} + (1-t) f_{o,exit}$
$f_{o,exit}$	0.5..2.0	dto.
f_s	0.1..1.0	
γ	-10.0..10.0	relative tilt angle (unit: degree)
f_{rect}	0.8..5.0	

Table 2: Parameter space of the NURBS-based secondary reflector shape for the test application

All of this sums up to the following 16 parameters that are varied within the listed limits during the adaptation of the secondary concentrator shape for the test application (see table 2).

6. Parameterization and setup of alternative shapes

In an effort to judge the optical performance of the designed NURBS-based secondary concentrator, alternative designs are also considered and compared to the NURBS solution. The obvious choice is the compound parabolic concentrator (CPC) suggested by Winston (1974) (see also Winston et al. (2005)).

As it proved not to be trivial to find a correct surface equation for this concentrator shape, the one used in this contribution is stated here (without derivation).

A parabola that forms one boundary arm of a CPC in two-dimensional space with half acceptance angle α and exit aperture radius a has the focal length f:

$$f = \frac{a}{2}(1+\sin\alpha). \tag{21}$$

When a coordinate z runs along the symmetry axis of the CPC, starting from the exit aperture plane, the radius $r(z)$ of the CPC is:

Name	Value range	Comment
α	20.0..65.0	max acceptance half angle (unit: deg)
$2a$	0.5..4.0	diameter of the exit aperture (unit: m)
γ	0.0..45.0	relative tilt angle (unit: degree)

Table 3: Parameter space of the CPC secondary reflector shape for the test application

$$
\begin{aligned}
r(z) &= \sqrt{x_0^2 + (y_0 - f)^2 - z^2}, \text{with} \qquad (22)\\
x_0^2 &= -8f\tan\alpha\sqrt{kfz + (\tan^2\alpha + 1)f^2} + 4kfz + 4(2\tan^2\alpha + 1)f^2\\
k &= \sin\alpha\tan\alpha + \cos\alpha\\
y_0 &= \frac{1}{4f}x_0^2.
\end{aligned}
$$

Introducing two coordinates (x, y) to describe the position within each cross section plane parallel to the exit aperture plane, one can replace r in eq. (22) using the identity $r = \sqrt{x^2 + y^2}$ which yields the surface equation.

Trying to find a CPC that best fits the test application, the parameters listed in table 3 have been varied within the given bounds. Much like the NURBS-based concentrator, the CPC has also been allowed to be tilted with respect to the receiver aperture plane. In both cases either the topmost or the lowest point of the secondary concentrator exit aperture has been required to touch the receiver. Whenever the relative tilt angles γ are non-zero, this leads to an intersection between the receiver plane and the secondary reflector. In that case, the area of this plane has been used as A_{rec} to calculate the emission losses (see eq.(20)). However, the relative tilt angle has never been allowed to grow such that the intersection plane would also have touched the entrance plane of the concentrator as this would have changed the topology of the concentrator from "tube" to "open mirror". Of course the same cost function as in the NURBS case has been applied.

In addition to the CPC, a receiver without any secondary concentrator has also been calculated, i.e. adapted to the test application scenario. The only parameter varied in that case was the diameter of the receiver aperture.

7. Adaptation and simulation results

Using the simulated annealing algorithm sketched in Press et al. (2002) (see §10.9 p. 448ff), the parameters listed in tables 2 and 3 have been varied within the given bounds and the resulting CPC- and NURBS-based secondary reflectors were used in the test application. The estimated annual yield as outlined in section 4 served as cost function. Both the ray tracing and the parameter search have been coded manually in Object Pascal and compiled to a multi-threaded native executable. About 1500 epochs were necessary for each

search run in the NURBS-based shape case and roughly 300000 rays were traced for each cost function evaluation. Consequently, one search run took in the order of several days on a single PC with two CPUs clocked with 3.2 GHz and hyperthreading turned on.

The best shapes found for both cases are listed in table 4 and depicted in fig. 3. Additionally, figures are given for the case when no secondary reflector at all is used in the test application.

As can be seen, the NURBS-based shape performs best. It leads to significantly more annual yield than the CPC secondary reflector. Observing the even lower yield estimated for the first case, it can be concluded that using a secondary reflector obviously is advantageous.

The concentration ratio is the entrance aperture area of the secondary reflector divided by the exit aperture area. "Max. semi-angle" is the maximum angle between an incoming ray (that eventually hits the receiver) and the entrance aperture normal of the secondary receiver. While the receiver area, i.e. the radiating area A_{rec}, is the same for both reflectors, the concentration ratio of the NURBS-based shape is higher than that of the CPC indicating that the NURBS-based shape collects more of the radiation emitted by the heliostats than the CPC. This is remarkable because the acceptance angle of the NURBS-based shape is also much higher than that of the CPC.

Put in numbers, the NURBS-based shape collects about 95% of all rays that leave the heliostats. And about 93.6% of these rays leave the concentrator through the exit aperture, i.e. get absorbed by the receiver. The 300000 rays that cross the entrance aperture lead to roughly 400000 reflections inside of the concentrator, giving an average of about 1.3 reflections per ray. Thereby more or less the entire reflective surface gets touched by a ray at least once.

8. Summary and conclusion

In this contribution it is shown how NURBS free form surfaces can be used to mathematically describe and model active, i.e. reflective surfaces in optical systems. Transforming the virtually unlimited variety of possible shapes into an application-specific subspace by means of a model with few parameters paves the

Case	**Receiver area**	**Conc. ratio**	**Max. semi-angle**	**Estimated annual yield**
No secondary reflector	1.54 m^2	1.0	-	1210 MWh
CPC	1.50 m^2	1.23	31.4°	1580 MWh
NURBS-based shape	1.50 $m2$	1.5	60.7°	2180 MWh

Table 4: Simulation results after adaptation

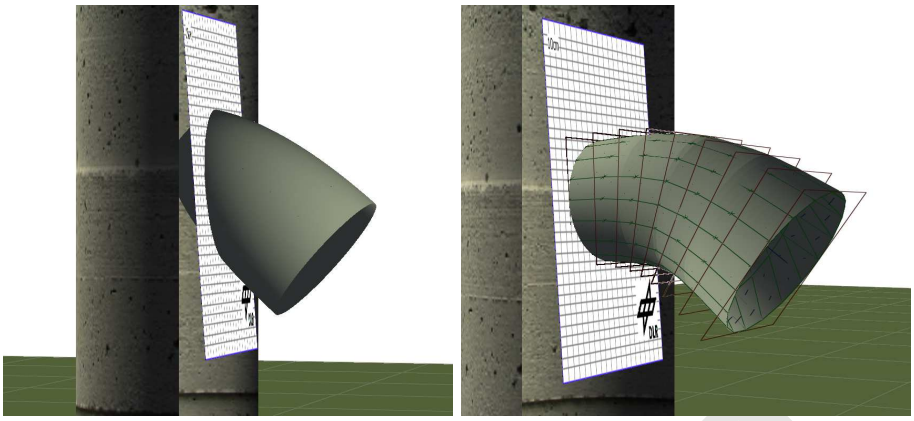

Figure 3: The adapted secondary concentrators: CPC (left) and NURBS (right)

way towards automated design of optical devices based on NURBS surfaces. It is proved by demonstration that this is feasible on modern personal computers.

For a specific test application, a secondary reflector based on a NURBS shape is designed and compared to a CPC secondary concentrator. It turned out that the specifically designed NURBS-based shape outperforms the CPC shape.

References

Abert, O., Geimer, M., Müller, S., September 2006. Direct and fast ray tracing of nurbs surfaces. In: Proceedings of the IEEE Symposium on Interactive Ray Tracing. Salt Lake City, UT, USA, pp. 161–168.

Kribus, A., Huleihil, M., Timinger, A., Ben-Mair, R., 2000. Performance of a rectangular secondary concentrator with an asymmetric heliostat field. Solar Energy 69 (2), 139–151.

Martin, W., Cohen, E., Fish, R., Shirley, P., 2000. Practical ray tracing of trimmed nurbs surfaces. Journal of graphics tools 5 (1), 27–52.

Nishita, T., Sederberg, T. W., Kakimoto, M., 1990. Ray tracing trimmed rational surface patches. Computer Graphics 24 (4), 337–345.

Piegl, L., Tiller, W., 1996. The NURBS book. Springer.

Press, W. H., Teukolsky, S. A., Vettering, W. T., Flannery, B. P., 2002. Numerical recipes in C++, 2nd Edition. Cambridge University Press, the art of scientific computing.

Timinger, A., Spirkl, W., Kribus, A., Ries, H., 2000. Optimized secondary concentrators for a partitioned central receiver system. Solar Energy 69 (2), 153–162.

Winston, R., 1974. Principles of a solar concentrator of a novel design. Solar Energy 16, 89–95.

Winston, R., Miñano, J. C., Benítez, P., 2005. Nonimaging optics. Elsevier Academic Press.