The Mathematics of Surfaces VIII
Edited by Robert Cripps
Published by Information Geometers, 1998
ISBN 1-874728-15-1
(213-231)

Interpolation and approximation with ruled surfaces

J. Hoschek and U. Schwanecke
Darmstadt University of Technology

Abstract

In this paper we introduce linear methods for interpolation or approximation of given data sets (scattered points, lines, planes) or explicitly given surfaces by ruled surfaces (non developable or developable). For surface description we will use the dual representation of ruled surfaces.

1 Introduction

A ruled surface is a surface which has the property that through every point of the surface there passes a straight line which lies entirely on the surface. Thus, the surface is covered by a one parameter set of straight lines, called rulings or generators. Each ruling lies in the tangent plane at every point of the ruling. If the tangent plane varies from point to point the ruled surface is called non developable (general ruled surface), the ruled surface is developable if and only if the tangent planes at all points of a ruling coincide to one plane. Thus the tangent planes of a developable surface form a family depending on one parameter, while the tangent planes of a general ruled surface form a family depending on two parameters.

The usual description of a ruled surface in the "Bézier B–Spline world" is a tensor–product Bézier or B–Spline surface linear in one parameter. This description has some disadvantages, especially since there are no criteria to decide easily whether a ruled surface is developable or not. Another approach was introduced in [15], in which the dual mapping with dual numbers, well known in kinematics, is used whereas in [2],[14] Pluecker coordinates are introduced for ruled surface description.

In [10],[11],[13] interpolation and approximation for developable surfaces is developed with the help of dual B–Splines [7]. In this paper we will introduce a general method for interpolation and approximation with ruled surfaces in dual description. The advantages of this approach are

- the developable surfaces are to be considered as special cases and can be treated with the same concept,
- one can interpolate and approximate lines, points and tangent planes linearly.

The usual B–Spline representation of a rational ruled surface in Euclidean 3–space $\mathbf{E}^3$ has the parametric representation

$$\mathbf{Y}(u,v) = \sum_{i=0}^{n} \mathbf{D}_{i0} N_{ik}(u)(1-v) + \sum_{i=0}^{n} \mathbf{D}_{i1} N_{ik}(u) v$$

in which $\mathbf{D}_{i0}, \mathbf{D}_{i1}$ are the control points in homogeneous coordinates and $N_{ik}(u)$ are normalised B–Spline functions of order k. If the $\mathbf{D}_{ij}$ are not at infinity we can write $\mathbf{D}_{ij} = (\omega_{ij}, \omega_{ij}x_{ij}, \omega_{ij}y_{ij}, \omega_{ij}z_{ij})$ with weights $\omega_{ij} \neq 0$ and $\mathbf{d}_{ij} = (x_{ij}, y_{ij}, z_{ij})$ as Cartesian coordinate vectors of the control points. We will only consider open rational B–Spline surfaces, thus we choose a knot vector $\mathbf{T} := (v_0 = v_1 = \dots = v_{k-1}, v_k, \dots, v_n, v_{n+1} = \dots = v_{n+k})$ with a monotone sequence v_i. If the knot sequence has only the values $\mathbf{T} := (v_0 = v_1 = \dots = v_{k-1}, v_k = v_{k+1} = \dots = v_{2k-1})$ the B–Spline description changes to the Bézier representation.

While $\mathbf{Y}(u,v)$ describes a surface as a focus of points, we can also interpret a ruled surface as an envelope of its tangent planes. For this transformation we use the **principal of duality** from projective geometry, therefore we have to exchange

points $\Longleftrightarrow$ **planes**

and obtain as a representation of a dual linear tensor–product B–Spline surface

$$\begin{aligned}\mathbf{Y}(u,v) &= \sum_{i=0}^{n} \mathbf{U}_{i0} N_{ik}(u)(1-v) + \sum_{i=0}^{n} \mathbf{U}_{i1} N_{ik}(u) v \\ &=: (y_0(u,v), y_1(u,v), y_2(u,v), y_3(u,v))\end{aligned} \tag{1.1}$$

in which the vectors $\mathbf{U}_{ij}$ are the homogeneous plane coordinate vectors of the control planes. Formula (1.1) represents a two parametric set of planes whose explicit equation in $\mathbf{E}^3$ is

$$y_0(u,v) + y_1(u,v)x + y_2(u,v)y + y_3(u,v)z = 0. \tag{1.2}$$

Then the required ruled surface is the envelope of the planes (1.2). This equation determines a one parametric set of planes for $v = const.$, whose envelope is a developable surface.

The parameter v in (1.1) is unimportant for the description of a ruled surface, it only gives information on a region of interest!

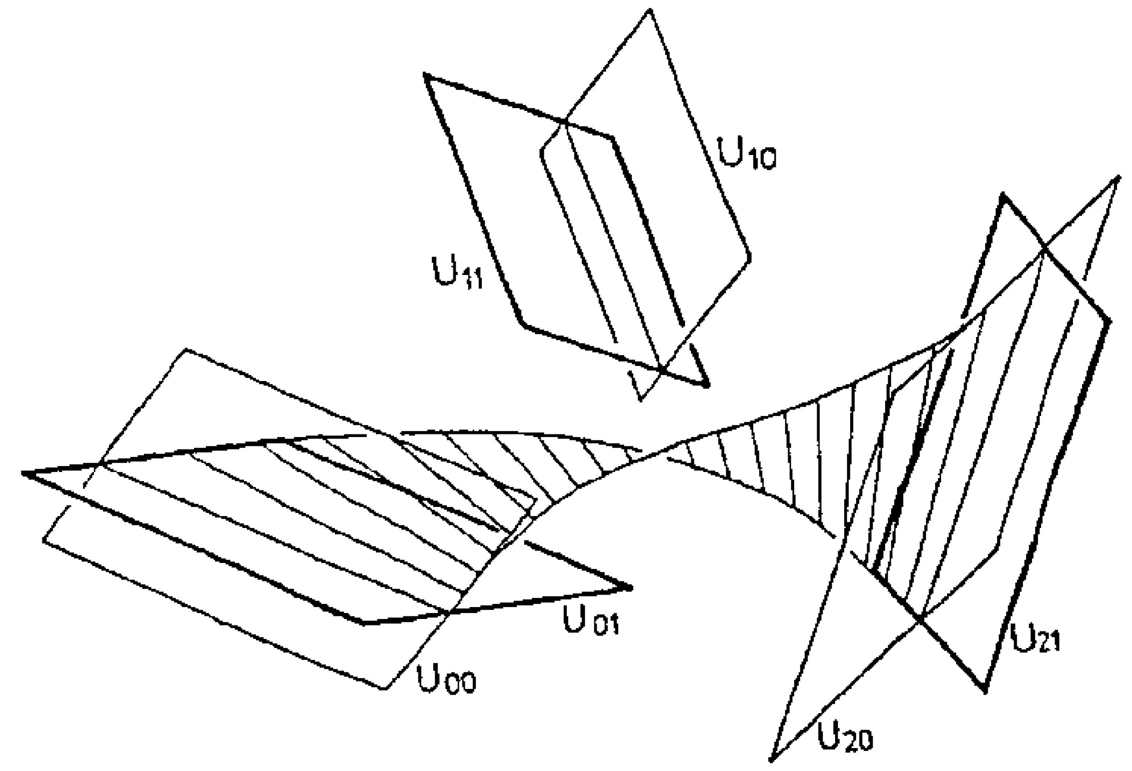

FIGURE 1. A (general) ruled surface and its control planes

Taking the well known properties of the B–Spline surfaces it follows for the control planes $\mathbf{U}_{ij}$, that $\mathbf{U}_{00}, \mathbf{U}_{01}, \mathbf{U}_{n0}, \mathbf{U}_{n1}$ are tangent planes at the boundary generators of the ruled surface over the given knot vector, the lines $\mathbf{U}_{00} \cap \mathbf{U}_{01}$ and $\mathbf{U}_{n0} \cap \mathbf{U}_{n1}$ determine the boundary generators of the ruled surface, additionally $\mathbf{U}_{00} \cap \mathbf{U}_{01} \cap \mathbf{U}_{10}, \mathbf{U}_{00} \cap \mathbf{U}_{01} \cap \mathbf{U}_{11}, \mathbf{U}_{n0} \cap \mathbf{U}_{n1} \cap \mathbf{U}_{n-1,0}, \mathbf{U}_{n0} \cap \mathbf{U}_{n1} \cap \mathbf{U}_{n-1,1}$ determine the corner points of the ruled surface patch (see Figure 1).

The generators of the ruled surface follow as an intersection of the plane $\mathbf{Y}(u,v)$ and its derivatives

$$\frac{\partial \mathbf{Y}}{\partial v}(u) = \mathbf{Y}_v(u)$$

and have the direction vector

$$\mathbf{a}(u,v) = \mathbf{N}(u,v) \wedge \mathbf{N}_v(u,v) \tag{1.3}$$

with $\mathbf{N}(u,v)$ as normal vector of the plane (1.2). A point on the ruled surface can be described by

$$\mathbf{P}(u,v) = \mathbf{Y}(u,v) \wedge \mathbf{Y}_v(u,v) \wedge \mathbf{Y}_u(u,v) \tag{1.4}$$

with $\mathbf{Y}_v$ and $\mathbf{Y}_u$ as derivatives of $\mathbf{Y}$ with respect to v and u.

For $\mathbf{Y}_v(u_0) = 0$ the generator at u_0 is a so-called torsal ruling. In this case the surface has locally the behaviour of a developable surface, the parametric representation $\mathbf{Y}$ does not depend on the parameter v. For torsal rulings the generators and points must be calculated according to (1.10) and (1.11) instead of (1.3) and (1.4).

With the help of (1.4) we can transform the dual representation into the usual tensor product representation in the point space. If we, formally, write (1.1)

$$\mathbf{Y}(u,v) := \mathbf{L}_0(u)(1-v) + \mathbf{L}_1(u)v \tag{1.5}$$

with

$$\mathbf{L}_0(u) = \sum_{i=0}^{n} \mathbf{U}_{i0} N_{ik}(u), \qquad \mathbf{L}_1(u) = \sum_{i=0}^{n} \mathbf{U}_{i1} N_{ik}(u),$$

we obtain from (1.4) as a tensor product representation

$$\mathbf{P}(u,v) = (\mathbf{L}_0 \wedge \mathbf{L}_1 \wedge \dot{\mathbf{L}}_0)(1-v) + (\mathbf{L}_0 \wedge \mathbf{L}_1 \wedge \dot{\mathbf{L}}_1)v \tag{1.6}$$

Thus, the corresponding tensor product surface has the order $(3k-2, 1)$.

The rulings $\mathbf{g}(u)$ of the general ruled surface are the intersection lines of the tangent planes $\mathbf{L}_0(u)$ and $\mathbf{L}_1(u)$. Instead of (1.3) we can use

$$\mathbf{g}(u) = \mathbf{L}_0(u) \cap \mathbf{L}_1(u).$$

Choosing $\mathbf{U}_{i0} = \mathbf{U}_{i1} =: \mathbf{U}_i$ in (1.1) the ruled surface is developable: (1.1) describes the one parametric set of planes.

$$\mathbf{Y}(u) = \sum_{i=0}^{n} \mathbf{U}_i N_{ik}(u). \tag{1.1a}$$

Formula (1.1a) now represents a one parametric set of planes with the explicit equation in $\mathbf{E}^3$

$$y_0(u) + y_1(u)x + y_2(u)y + y_3(u)z = 0. \tag{1.2a}$$

Developable surfaces can be unfolded or developed onto a plane without stretching or tearing. They are of considerable importance to sheet-metal-based industries. Those surfaces occur in many applications as with

windshield design, blank holder surfaces, for sheet metal forming process, aircraft skins, ship hulls, ductwork, feeder (shoulder surface of a packing machine) and within a lot more applications [3],[12] while general ruled surfaces are of large interest in architectural design [1], wire electric discharge machining [15],[16] and NC-milling with a cylindrical cutter [4].

An envelope of a one parametric set of planes is a developable surface and its edge of regression is the locus of the singularities. In general we want to create strips of developable B-Spline surfaces that are regular, i.e. containing no singular points. With the well-known properties of B-Spline curves [9] it follows that $\mathbf{U}_0$ and $\mathbf{U}_n$ are tangent planes at the boundaries of the developable surface over the given knot vector. The lines $\mathbf{U}_0 \cap \mathbf{U}_1$ and $\mathbf{U}_n \cap \mathbf{U}_{n-1}$ determine the boundary generators of the developable surface so $\mathbf{U}_0 \cap \mathbf{U}_1 \cap \mathbf{U}_2$ and $\mathbf{U}_n \cap \mathbf{U}_{n-1} \cap \mathbf{U}_{n-2}$ determine the points of regression at the boundary generators (see Figure 2). Further, it can be shown that a dual B-Spline curve is formed of developable Bézier surfaces pieced together along the rulings to the knots $t = v_l$ of multiplicity μ_l with the continuity $C^{k-\mu_l-1}$ [13].

The generators of the developable surface follow as an intersection of the plane $\mathbf{Y}(u)$ and its derivatives $\dot{\mathbf{Y}}(u)$ and have the direction vectors

$$\mathbf{a}(u) = \mathbf{N}(u) \wedge \dot{\mathbf{N}}(u) \tag{1.7}$$

with $\mathbf{N}(u)$ as normal vector of the plane (1.2a). The edge of regression or cuspoidal edge is obtained as an intersection $\mathbf{Y}(u) \cap \dot{\mathbf{Y}}(u) \cap \ddot{\mathbf{Y}}(u)$. Its explicit representation can be described by

$$\mathrm{C}(u) = \mathbf{Y}(u) \wedge \dot{\mathbf{Y}}(u) \wedge \ddot{\mathbf{Y}}(u). \tag{1.8}$$

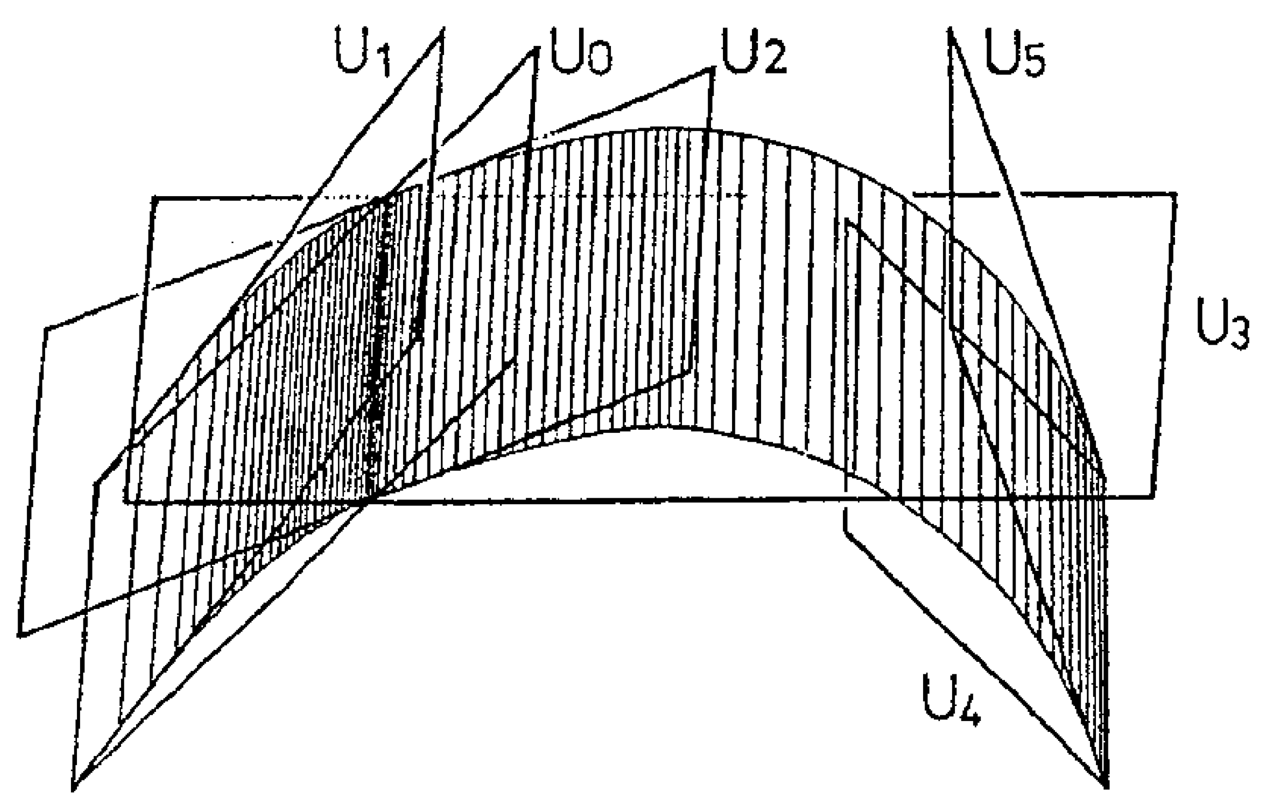

FIGURE 2. A developable surface and its control planes

Algorithms converting the dual representation of developable surfaces to a standard tensor product form have already been given in [13]. They are particularly simple if the patch is to be confined by planar boundary curves. This is due to the fact that the intersection of a developable NURBS surface $\mathbf{Y}$ with a plane $\mathbf{U}$ is a NURBS curve: its lines (of its dual form) are simply the intersections of the control planes of $\mathbf{Y}$ with the plane $\mathbf{U}$.

2 Interpolation with ruled surfaces

In order to avoid non-linearity we use a linear description of lines: if two plane vectors $\mathbf{V}_1$ and $\mathbf{V}_2$ intersect in a line $\mathbf{g} = \mathbf{V}_1 \cap \mathbf{V}_2$, then these planes determine a pencil of planes through $\mathbf{g}$. Each plane $\mathbf{V}$ of the pencil can be described with the help of real parameters λ_1, λ_2 by a linear combination

$$\mathbf{V} = \lambda_1 \mathbf{V}_1 + \lambda_2 \mathbf{V}_2. \tag{2.1}$$

First we will discuss interpolation by non developable or developable surfaces and discuss the following problems.

a) determine a ruled surface interpolating the generators

$$\mathbf{g}_j \ (j = 1 \dots M_1),$$

b) determine a ruled surface interpolating the boundary generators, the tangent planes in the corner points of a patch and some generators $\mathbf{g}_j$ in the interior of the surface ($j = 1 \dots M_2$),

c) determine a ruled surface interpolating some points $\mathbf{P}_j$ ($j = 1 \dots M_3$) and some generators $\mathbf{g}_j$ ($j = 1 \dots M_4$).

For these problems we get the following solutions for **non developable** surfaces:

For **problem a)** we can subdivide our problem in two parts: The generators of a ruled surface according to (1.1), (1.5) are determined by the intersection of the planes $\mathbf{L}_0(u), \mathbf{L}_1(u)$.

Each given generator may be determined by the intersection of the planes $\mathbf{E}_j, \mathbf{Q}_j$ ($j = 1 \dots M_1$) with the parameter value $u = u_j$. Thus, we have to solve the two linear systems

$$\mathbf{L}_0(u_j) = \alpha_j \mathbf{E}_j + \mathbf{Q}_j, \quad \mathbf{L}_1(u_j) = \mathbf{E}_j + \beta_j \mathbf{Q}_j \tag{2.2}$$

with α_j and β_j as scalar unknowns.
In general, we could also introduce a scalar factor for $\mathbf{Q}_j$ on the left hand side equation and for $\mathbf{E}_j$ on the right hand side equation. Our special choice of scalar factors leads to a better numerical condition of the matrix solving the interpolation problem.

In (1.5) and (2.2) we have two sets of independent equations, thus we can decompose our interpolation problem in two parts: the left hand side equations in (1.5) and (2.2) and the right hand side equations. For each problem we obtain from (2.2) $4M_1$ equations with $4(n+1)$ components of the control planes and M_1 scalar unknowns α_j (or β_j). Thus, we have to fulfil the balance equation

$$4n - 3M_1 + 4 = 0. \tag{2.3}$$

For **problem b)** we have $\mathbf{U}_{00} = \mathbf{T}_{00}, \mathbf{U}_{01} = \mathbf{T}_{01}, \mathbf{U}_{n0} = \tau_{n0}\mathbf{T}_{n0}, \mathbf{U}_{n1} = \tau_{n1}\mathbf{T}_{n1}$ with $\mathbf{T}_{ik}$ as tangent planes at the corner points and τ_{n0}, τ_{n1} as scalar factors. The corner points $\mathbf{P}_{ik}$ must satisfy the equations of the tangent planes $\mathbf{T}_{ik}$ trivially but we have, furthermore, the conditions

$$\mathbf{U}_{10} \cdot \mathbf{P}_{00} = 0, \ \mathbf{U}_{11} \cdot \mathbf{P}_{01} = 0, \ \mathbf{U}_{n-1,0} \cdot \mathbf{P}_{n0} = 0, \ \mathbf{U}_{n-1,1} \cdot \mathbf{P}_{n1} = 0 \tag{2.4}$$

in which the corner points $\mathbf{P}_{ik}$ are used in a homogeneous representation.

Equations of type (2.2) have to hold for the generators in the interior of this surface. Again, we can separate the interpolation problem in two parts determined by the left hand side equations in (2.4) and (2.2) and the right hand side equations. In each sub-problem we have the following numbers of unknowns: $4(n-1)$ components of the control planes, one unknown τ_{n0} (or τ_{n1}) and M_2 unknowns α_j (or β_j) according to (2.2).

We obtain $4M_2$ equations from (2.2) and additionally 2 equations from condition (2.4). Thus, we have as a balance condition

$$4n - 5 = 3M_2 \tag{2.5}$$

For **problem c)** we have to fulfil the point conditions

$$\mathbf{P}_j \cdot \mathbf{Y}(u_j, v_j) = 0, \ \mathbf{P}_j \cdot \mathbf{Y}_u(u_j, v_j) = 0, \ \mathbf{P}_j \cdot \mathbf{Y}_v(u_j, v_j) = 0 \tag{2.6}$$

Parameters v_j can be arbitrary chosen. Therefore, it is not possible to separate the problem in two parts: the conditions (1.5), (2.2) for interpolating of lines have to be fulfilled simultaneously. We have as unknowns the $8(n+1)$ components of the control planes and $2M_4$ scalar unknowns α_j, β_j. From M_3 points we get $3M_3$ equations and from M_4 generators we get $8M_4$ equations. This leads to the balance condition

$$6M_4 + 3M_3 = 8(n+1) \tag{2.7}$$

Figure 3 contains an example of an interpolating ruled B-spline surface. We have chosen 6 points $\mathbf{P}_i$ and 9 generators $\mathbf{g}_j$ and order $k = 3$ with $n = 8$.

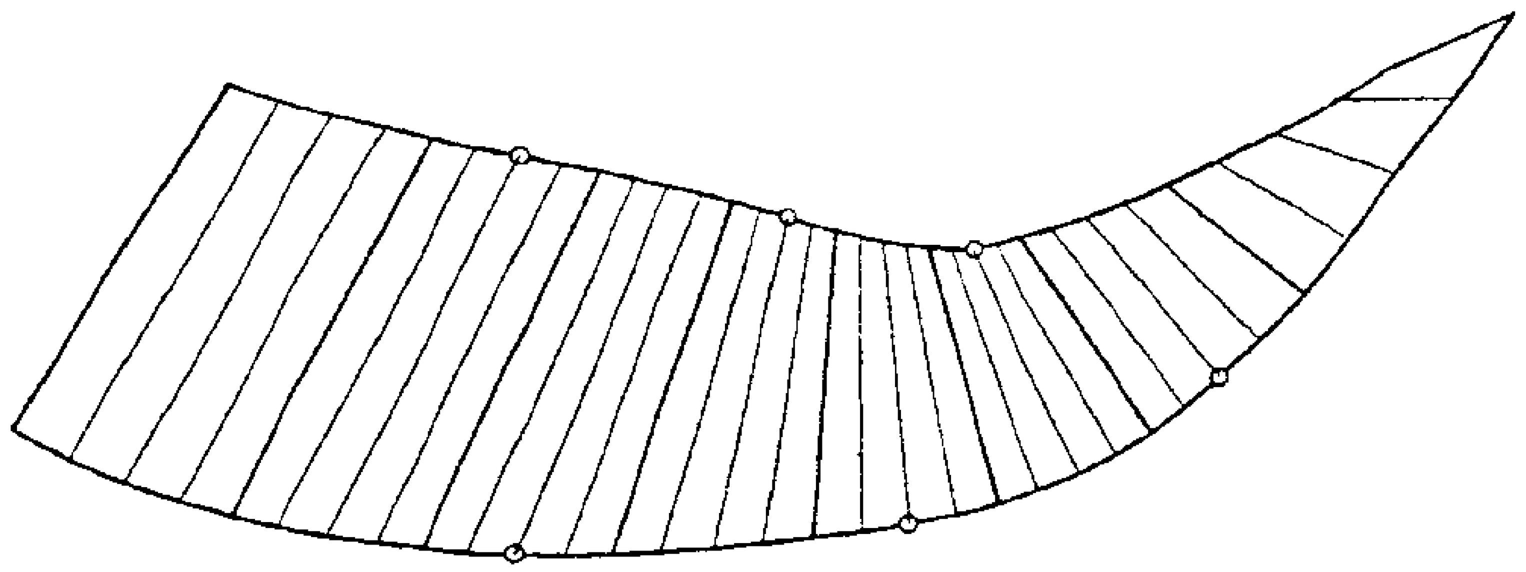

FIGURE 3. Interpolating ruled surface with given points (circles) and generators (bold)

Now, we will discuss the interpolation with **developable surfaces** and consider the problems **a**), **b**), **c**) again:

For **problem a**) we have instead of (1.5), (2.2) with (1.1a) the conditions

$$\sum_{i=0}^{n} \mathbf{U}_i N_{ik}(u_j) = \alpha_j \mathbf{E}_j + \mathbf{Q}_j, \qquad \sum_{i=0}^{n} \mathbf{U}_i \dot{N}_{ik}(u_j) = \mathbf{E}_j + \beta_j \mathbf{Q}_j \tag{2.8}$$

with α_j, β_j as scalar unknowns. Both equations in (2.8) are dependent on the unknown control planes $\mathbf{U}_i$. Therefore we have to solve the corresponding system simultaneously: For M_1 given lines we obtain $8M_1$ equations with $4(n+1)$ components of the control planes and $2M_1$ scalar unknowns α_j, β_j. Instead of (2.3) we get the balance condition

$$4n - 6M_1 + 4 = 0.$$

For **problem b**) we now have $\mathbf{U}_0 = \mathbf{T}_0$, $\mathbf{U}_n = \tau_n \mathbf{T}_n$ as tangent planes at the boundary rulings. Instead of (2.4) we now have

$$\mathbf{U}_1 \cdot \mathbf{P}_0 = 0, \qquad \mathbf{U}_{n-1} \cdot \mathbf{P}_n = 0 \tag{2.9}$$

and obtain for M_2 given rulings $8M_2$ equations and two additional equations from (2.9). As unknowns we have $4(n-1)$ components of the control planes and $2M_2$ unknowns α_j, β_j. This leads to the balance condition

$$4n - 6 = 6M_2.$$

For **problem c**) we have to fulfil the point conditions

$$\mathbf{P}_j \cdot \mathbf{Y}(u_j) = 0, \qquad \mathbf{P}_j \cdot \mathbf{Y}_u(u_j) = 0$$

and get for M_3 given points $2M_3$ equations and for M_4 given rulings $8M_4$ equations. On the other hand we have $4(n+1)$ unknown components of the control planes and $2M_4$ unknown parameters α_j, β_j. This leads to the balance condition

$$2M_3 + 6M_4 = 4(n+1).$$

Figure 4 gives an example of an interpolating developable dual B-Spline surface of order 8 (6 generators (**problem a**) are given):

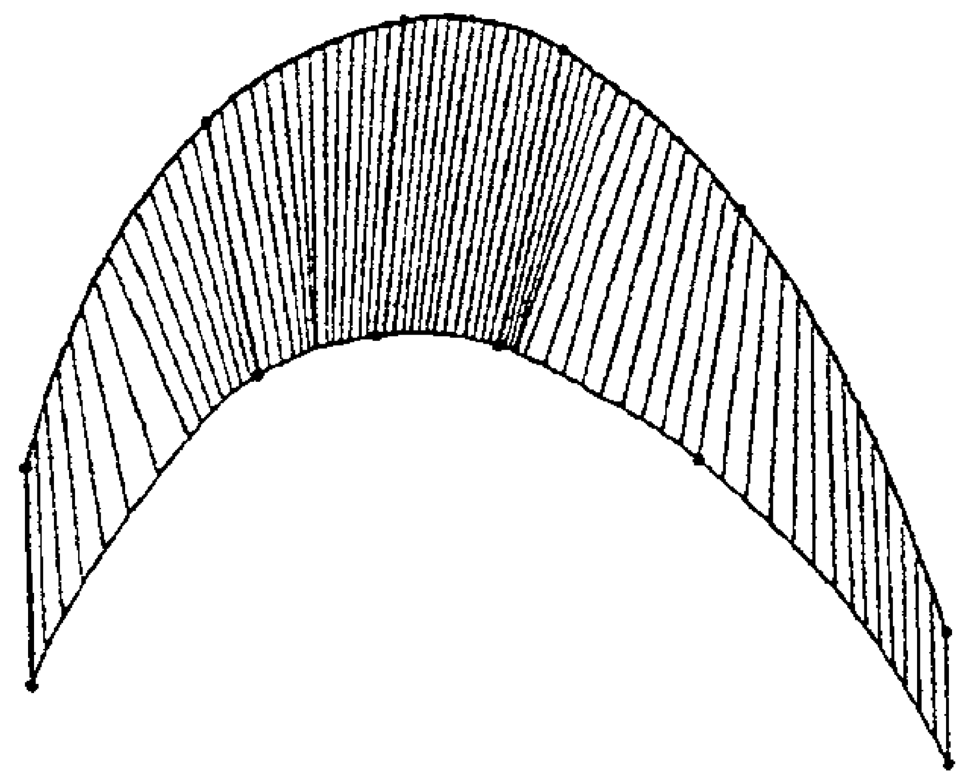

FIGURE 4. Interpolating developable surface (given generators bold)

3 Approximation with ruled surfaces

In this section we will consider the following two problems:

Problem A): given an analytic surface or tensor product surface, we require a ruled surface whose generators g_j are as close as possible to the given surface.

Problem B): given a set of scattered data points $\mathbf{P}_j$ possibly from a laser scan, we require a ruled surface whose generators g_j are as close as possible to the given points.

The given surface or the set of scattered data points determine the area of interest (a suitable box around the surface of the data).

The crucial point of approximation with ruled surfaces is the choice of an appropriate error measurement. In [14] Pluecker coordinates are used for error measurement, while in [11] the distance from a point to a line was used, which leads unfortunately to a non linear optimisation problem. In the presented paper we will introduce an error measurement between planes ([10]) which leads to **linear** algorithms. We only require the assumption that every tangent plane of the given surface must intersect the z-axis of our coordinate system!

For **problem A)** we determine on the given surface two sets of tangent planes $\mathbf{E}_j$, $\mathbf{Q}_j$ along two curves $v = const.$ at points with the parameter values u_j $(j = 0, ..., M)$. With our assumption on the coordinate system the tangent planes can have the explicit form

$$z = e_0 + e_1 x + e_2 y \qquad \text{or}$$

$$\begin{aligned} \mathbf{E}_j(x,y) : &= e_j^0 + e_j^1 x + e_j^2 y \\ & \qquad\qquad\qquad (j = 0, ..., M) \\ \mathbf{Q}_j(x,y) : &= q_j^0 + q_j^1 x + q_j^2 y \end{aligned} \tag{3.1}$$

Both sets of planes $\mathbf{E}_j$, $\mathbf{Q}_j$ will be approximated by two sets of tangent planes (two dual B-Spline curves) $\mathbf{L}_0(u)$ and $\mathbf{L}_1(u)$ according to (1.5) and we choose as objective functions

$$\sum_{j=0}^{M} ||\mathbf{L}_0(u_j) - \mathbf{E}_j||^2 \longrightarrow min., \qquad \sum_{j=0}^{M} ||\mathbf{L}_1(u_j) - \mathbf{Q}_j||^2 \longrightarrow min. \tag{3.2}$$

In our coordinate system we can normalise the control planes $\mathbf{U}_{is}$ $(s = 0, 1; i = 0, ..., n)$ by

$$\mathbf{U}_{is} = (u_{is}^0, u_{is}^1, u_{is}^2, u_{is}^3)^\top = (\omega_{is}^0, \omega_{is}^1, \omega_{is}^2, -1)^\top, \tag{3.3}$$

thus the plane equations $\mathbf{L}_0(u)$, $\mathbf{L}_1(u)$ have, with respect to (1.1) and (1.5), the explicit form

$$\mathbf{L}_0(u_j)(x,y) := l_0^0(u_j) + l_0^1(u_j)x + l_0^2(u_j)y$$

$$\mathbf{L}_1(u_j)(x,y) := l_1^0(u_j) + l_1^1(u_j)x + l_1^2(u_j)y.$$

Using the L^2-norm for our minimisation problem we obtain from (3.2)

$$\begin{aligned} I_0 &:= \sum_{j=0}^{M} \int_c^d \int_a^b (\mathbf{L}_0(u_j)(x,y) - \mathbf{E}_j(x,y))^2 dx dy \longrightarrow min. \\ I_1 &:= \sum_{j=0}^{M} \int_c^d \int_a^b (\mathbf{L}_1(u_j)(x,y) - \mathbf{Q}_j(x,y))^2 dx dy \longrightarrow min. \end{aligned} \tag{3.4}$$

with $[a,b] \times [c,d]$ as domain of our interest over the (x,y)-plane. In geometric terms we will minimise the z-distances between the tangent planes of a given surface and the approximation surface over the domain $[a,b] \times [c,d]$.

The minimum of (3.4) is determined by differentiation of (3.4) with respect to the unknown components ω_{is}^0, ω_{is}^1, ω_{is}^2 of the control points according to (3.3):

$$\frac{\partial I_s}{\partial \omega_{is}^0} = \frac{\partial I_s}{\partial \omega_{is}^1} = \frac{\partial I_s}{\partial \omega_{is}^2} = 0 \qquad (s = 0, 1;\ i = 0, ..., n). \tag{3.5}$$

We introduce the matrix

$$G := \begin{pmatrix} 2 & a+b & d+c \\ 6(a+b) & 4(a^2+b^2+ab) & 3(a+b)(c+d) \\ 6(d+c) & 3(a+b)(c+d) & 4(c^2+d^2+cd) \end{pmatrix}, \tag{3.6}$$

and the abbreviation (with $l = 0, ..., n;\quad i = 0, ..., n$)

$$n_{li} := \sum_{j=0}^{M} N_{lk}(u_j) N_{ik}(u_j) \tag{3.7}$$

and obtain after integration of (3.4) from (3.5) for $s = 0$ the $3(n+1) \times 3(n+1)$ linear system with the unknown components $\omega_{i0}^0, \omega_{i0}^1, \omega_{i0}^2$

$$\sum_{j=0}^{M} G \begin{pmatrix} e_j^0 \\ e_j^1 \\ e_j^2 \end{pmatrix} N_{ik}(u_j) = \sum_{l=0}^{n} G \begin{pmatrix} \omega_{i0}^0 \\ \omega_{i0}^1 \\ \omega_{i0}^2 \end{pmatrix} n_{il} \qquad (i = 0, ..., n). \tag{3.8}$$

For $s = 1$ we have to exchange in (3.8) e_j^l by q_j^l and ω_{i0}^l by ω_{i1}^l $(l = 0, 1, 2)$. Because $\det(G) = 2(a-b)^2(c-d)^2 \neq 0$ for $a \neq b, c \neq d$, we can eliminate the matrix G in (3.8), thus the required solution is independent on the chosen domain.

We have to solve (3.8) for $s = 0, 1$ and obtain the control planes of the required approximation surface according to (3.3) and the surface representation with (1.1) or (1.5). Figure 5 gives an example of the approximation of a given surface by a ruled B-Spline surface of order 3 and $n = 20$. For the approximation 50 tangent planes were chosen at each boundary curve of the given surface.

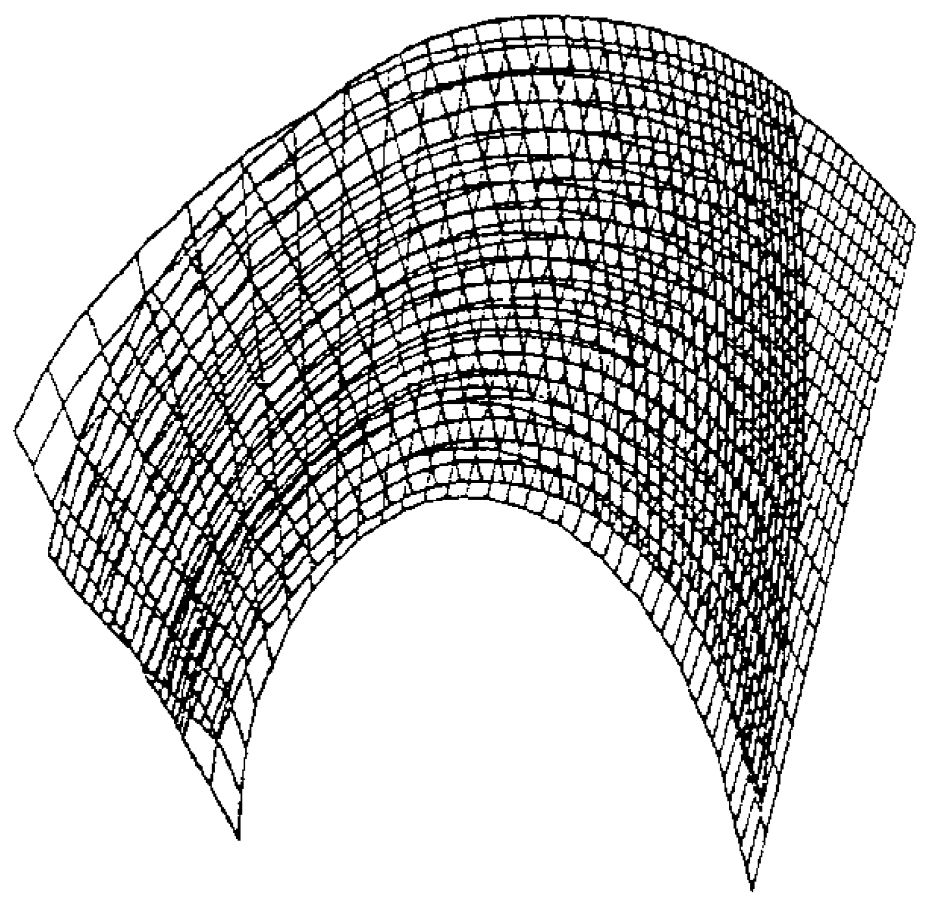

FIGURE 5. Approximation of a given surface (bold) by a ruled surface

If we approximate the given surface by a developable surface we have to solve only one system in (3.8) and find the solution according to (1.1a).

Now we will discuss our **problem B**) and we demonstrate here only the approximation by a developable surface: we use as a basis of our algorithm an approximating triangulation of the given set of points following an algorithm of Hoppe [5], where the facets of the triangulation approximate the set of points within an error tolerance δ.

For parametrisation of the triangulation we determine for each triangle the centre of gravity $\mathbf{C}_i$ and project these centres in the x, y-plane. The boundary of our region of interest may be determined by the minimal and maximal value of these projections (x_{min}, x_{max}, y_{min}, y_{max}). We choose the interval $I = [y_{min}, y_{max}]$ as parameter interval of the required approximation surface and subdivide I into $N + 1$ strips of the length σ. In a strip with index j ($j = 0, ..., N$) we pick out all triangles $\{\Delta_i\}$ with y-component of $\mathbf{C}_i$ in this strip, and determine a least square plane $\mathbf{E}_j$

by the mean values of the normals of $\{\Delta_i\}$ and the mean value of $\{\mathbf{C}_i\}$. The y-component c_j of the mean values of $\{\mathbf{C}_i\}$ is chosen as a parameter value of $\mathbf{E}_j$ by

$$u_j = \frac{c_j - y_{min}}{y_{max} - y_{min}}. \tag{3.9}$$

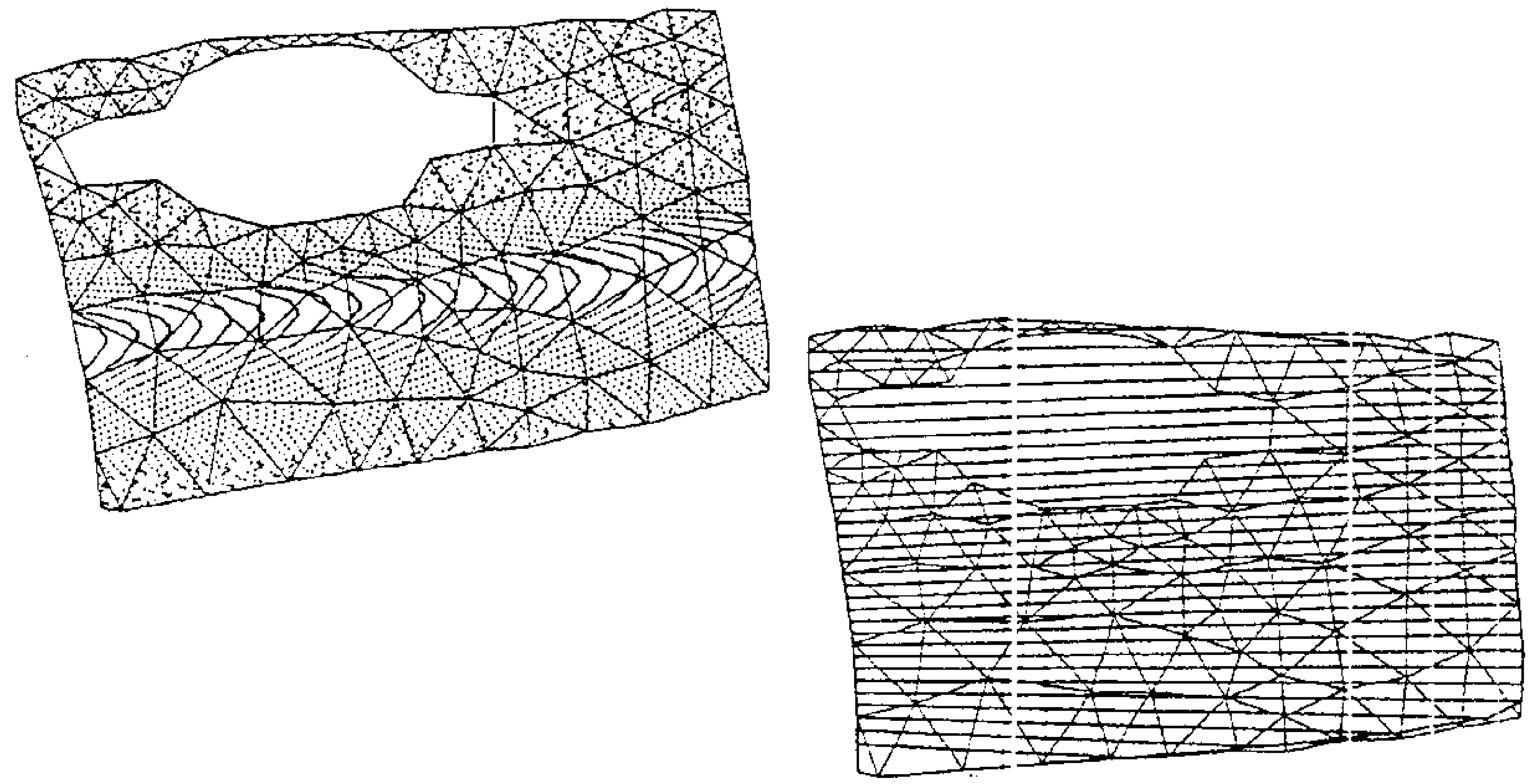

FIGURE 6. Set of scattered points and the corresponding triangulation (first figure) and generators of the approximating developable surface (second figure)

Analogously to (3.4) we will generate our required developable surface $\mathbf{Y}$ using the L^2 norm

$$\sum_{j=0}^{N} \int_{y_{min}}^{y_{max}} \int_{x_{min}}^{x_{max}} (\mathbf{Y}(u_j)(x,y) - \mathbf{E}_j(x,y))^2 dx dy \longrightarrow min. \tag{3.10}$$

Similar to problem A we have to solve the linear system (3.8) only for one index s and obtain with (3.3) and (1.1a) the required developable surface. Figure 6 demonstrates the approximation of a set of scattered points by a developable surface: in the first figure we have the cloud of points and the triangulation while the second figure shows a set of generators of an approximating developable Bézier surface of degree 4 with a prescribed error tolerance.

4 Approximation of surfaces of revolution

Approximations of surfaces of revolution by ruled surfaces are required for wire electric discharge machining. We will develop methods for approximation of surfaces of revolution by developable and (general) ruled surfaces. The approximation by developable surfaces can be used additionally in order to construct a surface of revolution with the help of planar segments (part of cones).

A surface of revolution is obtained by rotating a planar (or a non planar) curve about an axis of revolution rigidly connected with the curve through a complete revolution. We shall choose the axis of revolution as the z-axis of the coordinate system. The intersections of the surface of revolution with planes through the axis are called meridians. They are congruent curves and can be represented by the parametric (Bézier) representation (meridian curve in plane $x = 0$)

$$\mathbf{M}(u) = \sum_{i=0}^{n} \mathbf{b}_i B_i^n(u) = \begin{pmatrix} 0 \\ y(t) \\ z(t) \end{pmatrix} \qquad t \in [0,1] \qquad (4.1a)$$

with the Bernstein polynomials $B_i^n(u)$ of degree n and the control points

$$\mathbf{b}_i = (0, a_{i2}, a_{i3})^T \qquad (4.1b)$$

Whilst rotating around the z–axis a point of (4.1a) with the ordinate $y(t)$ describes the half circle (in the plane $z = 0$) [9]

$$R(v) = \begin{pmatrix} 1 \\ 0 \\ y(t) \end{pmatrix} B_0^2(v) + \begin{pmatrix} 0 \\ y(t) \\ 0 \end{pmatrix} B_1^2(v) + \begin{pmatrix} 1 \\ 0 \\ -y(t) \end{pmatrix} B_2^2(v) \qquad (4.2)$$

Combining (4.1) and (4.2) leads to the parametric representation of a surface of revolution as a rational tensor product surface

$$\mathbf{X}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{2} \mathbf{b}_{ij} B_i^n(u) B_j^2(v) \qquad (4.3)$$

with

$$\mathbf{b}_{i0} = \begin{pmatrix} 1 \\ 0 \\ a_{i2} \\ a_{i3} \end{pmatrix}, \mathbf{b}_{i1} = \begin{pmatrix} 0 \\ a_{i2} \\ 0 \\ 0 \end{pmatrix}, \mathbf{b}_{i2} = \begin{pmatrix} 1 \\ 0 \\ -a_{i2} \\ a_{i3} \end{pmatrix},$$

where $\mathbf{b}_{i1}$ denotes a Bézier point at infinity (only the direction $(a_{i2}, 0, 0)$ is active) [9].

First, we will approximate a given Bézier surface of revolution by a set of cones (see [8]):

For this purpose we need a linear approximation of the meridian. This is obtained by moving the end point Q of a line segment on the offset curves of the given meridian with a given error tolerance $\pm\varepsilon$ as offset, until one of the interior points of the line segment has the maximal distance ε_m from the meridian (see Figure 8). The set of line segments can be interpreted as a linear B–Spline curve, thus the representation of the corresponding surface of revolution can be analogously to (4.3). Geometrically speaking the surface consists of a set of cones.

The distances of the interior points of the line segments to the given meridian are determined by a local Newton method used also in reparametrisation of approximation curves (see [9]).

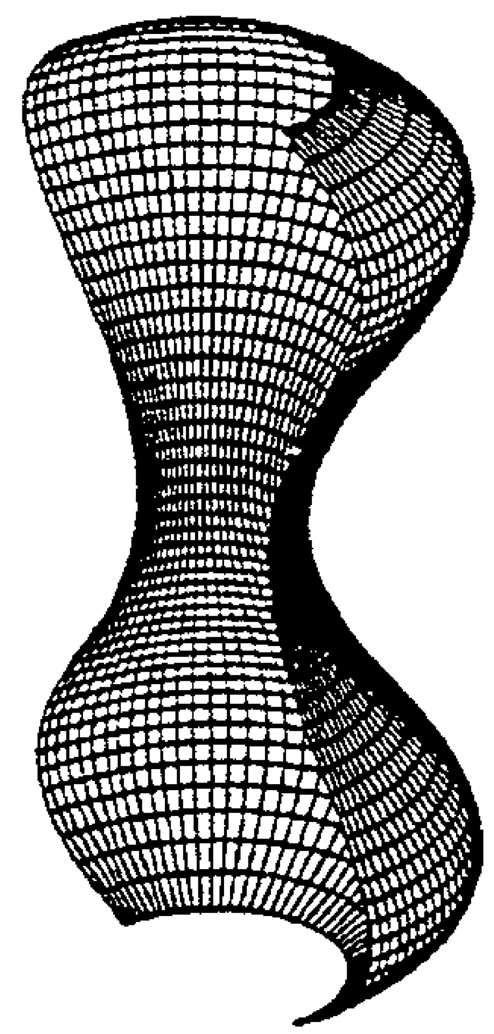

FIGURE 7. Given surface of revolution

We will show the results of our development with help of an example: given is a surface of revolution of a quartic meridian in Bézier representation with the control points (see Figure 7)

$$\mathbf{b}_0 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{b}_1 = \begin{pmatrix} 0 \\ 5 \\ 2 \end{pmatrix}, \mathbf{b}_2 = \begin{pmatrix} 0 \\ -5 \\ 4 \end{pmatrix}, \mathbf{b}_3 = \begin{pmatrix} 0 \\ 5 \\ 6 \end{pmatrix}, \mathbf{b}_4 = \begin{pmatrix} 0 \\ 1 \\ 8 \end{pmatrix}.$$

Figure 8 contains the approximation of the meridian of the surface in Figure 7 by linear segments, Figure 9 the corresponding approximating cones.

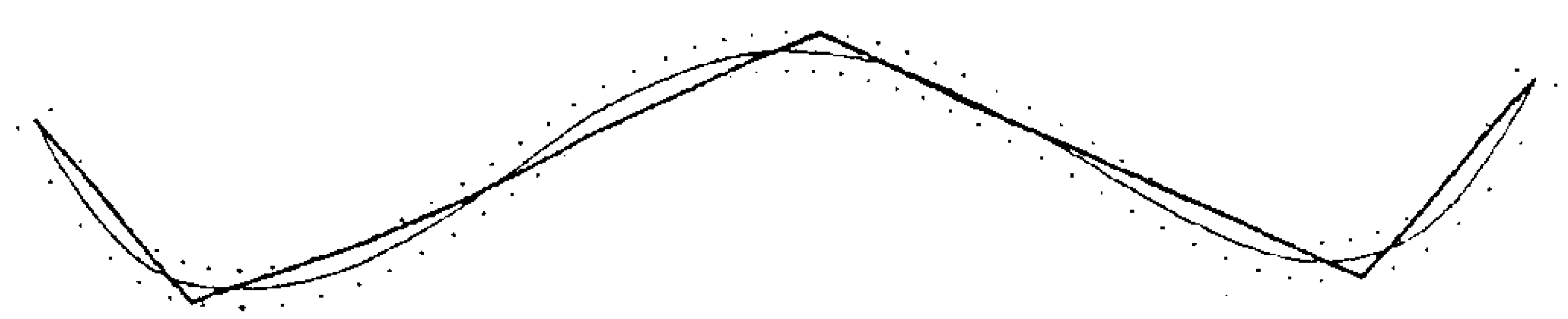

FIGURE 8. Approximation of the meridian from the surface in Figure 7 by line–segments with the given error tolerance 0.1mm. The offset–curves are dotted

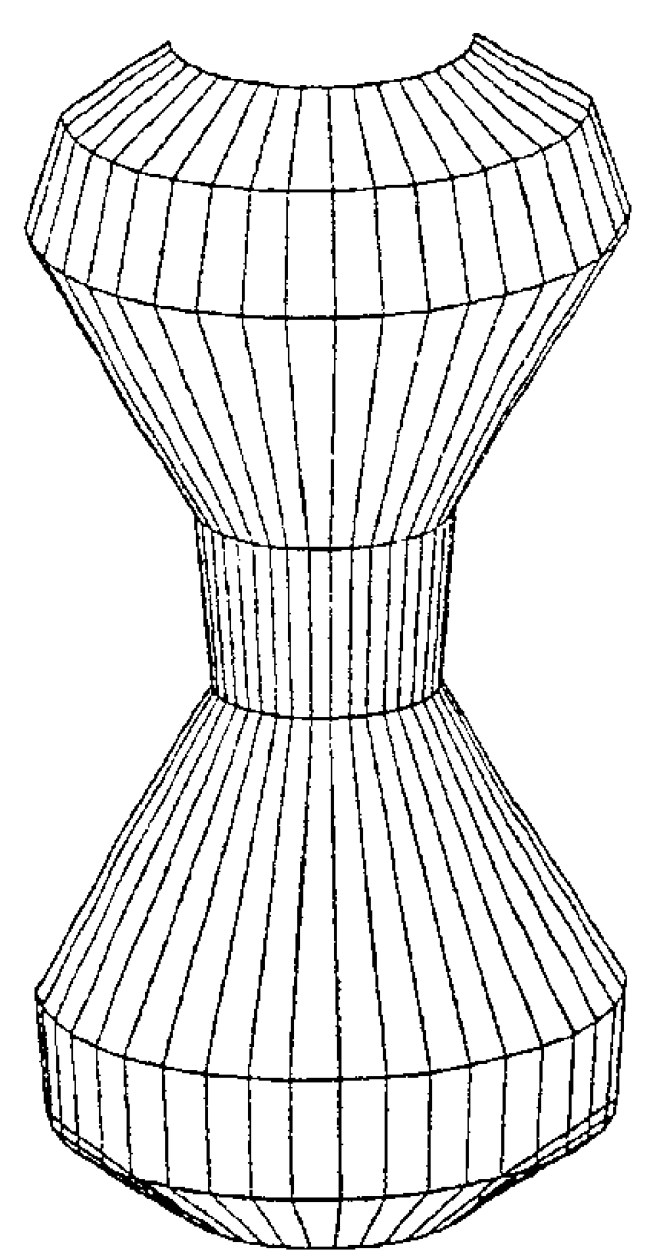

FIGURE 9. Conical approximation of the surface in Figure 7 ($\varepsilon = 0.05$)

Table 1 shows the dependency of the number s of segments and the error tolerance ε for the example in Figure 9.

ε	0.1	0.05	0.02	0.01
s	4	7	11	17

Table 1

For wire electric discharge machining approximations other than only by developable surfaces are used. Therefore we can approximate parts with negative Gaussian curvature by a hyperboloid of revolution which is generated by rotating a skew line around the axes of revolution. The meridian is a hyperbola. Thus, we can approximate a given surface in the parts with positive Gaussian curvature by cones and in regions with dominating negative Gaussian curvature by hyperboloid of revolution.

We use the parametric representation of a hyperboloid of revolution

$$\mathbf{X}(u, z_0) = (r\cos u, r\sin u, 0) + z_0(-\cot\gamma\sin u, \cot\gamma\cos u, 1), \qquad (4.4)$$

additionally the hyperboloid must be moved in z-direction by the translation vector $\mathbf{T} = (0, 0, m_0)$. We get the meridian of such a hyperboloid by intersecting the surface with the plane $x = 0$ and obtain the Cartesian equation $z = m_0 \pm \tan\gamma\sqrt{y^2 - r^2}$ (hyperbola) by eliminating the parameters. Analogously to the line segment approximation we move the two boundary points of the hyperbola segment on the offset curve of the meridian with an offset ε and minimise with the help of a Newton iteration for the parameters r, γ, m_0 the error distance between the given curve and the approximation curve. For the error measurement we use the dual description of (4.4) (maximal distance between parallel tangents).

Figure 10 contains the meridian of our example and the approximation with line and hyperbola segments and Figure 11 the approximating surface.

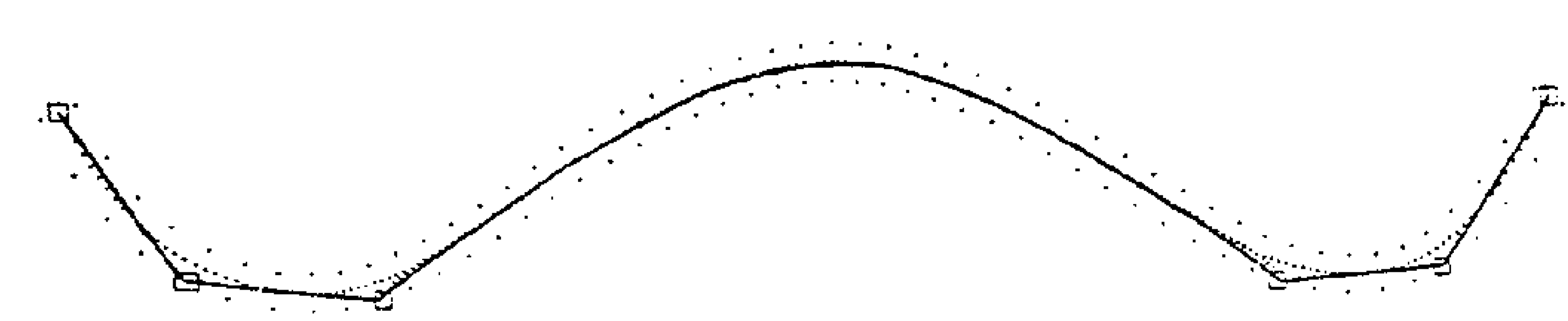

FIGURE 10. Approximation of the meridian of the surface of revolution in Figure 7 by line and hyperbolic segment

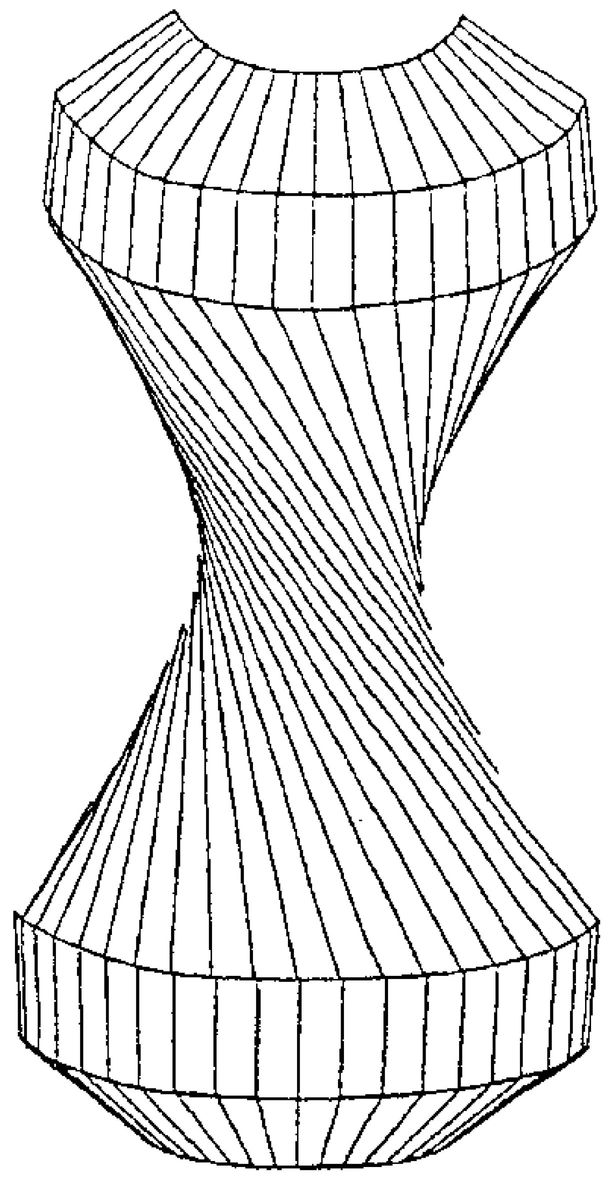

FIGURE 11. Approximation of a surface of revolution by cones and a hyperboloid

5 Conclusion

The proposed algorithms for interpolation and approximation of given points, lines, planes and surfaces by ruled surfaces are all linear and therefore very fast. The only problem is finding an appropriate parametrisation. With inappropriate parametrisations singularities can occur on the interpolation and approximation surfaces in the required region of interest.

References

[1] H. Brauner and W. Kickinger. *Baugeometrie*, 1, Bauverlag, 1977.

[2] H.-Y. Chen and H. Pottmann. Approximation by ruled Surfaces. Technical Report Nr. 46, Institut für Geometrie TU Wien, 1997.

[3] W.H. Frey and D. Bindschadler. Computer aided design of class of developable Bézier surfaces. General Motors R & D Publication 8057, 1992.

[4] Q.J. Ge. Kinematics-driven geometric modeling: a framework for simultaneous NC tool-path generation and sculptured surface design. *Proc. IEEE Intl. Conf. on Robotics and Automation*, Minneapolis, MN , 1819-1824, 1996.

[5] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics SIGGRAPH'92 Proceedings*, 26(2), 71–78, 1992.

[6] J. Hoschek. Liniengeometrie. Bibliographisches Institut, Mannheim, 1971.

[7] J. Hoschek. Dual Bézier curves and surfaces. In *Surfaces in Computer Aided Geometric Design*, Eds. R. E. Barnhill and W. Boehm, North Holland, 147-156, 1983.

[8] J. Hoschek. Approximation of surfaces of revolution by developable surfaces, will be published in Computer Aided Design.

[9] J. Hoschek and D. Lasser.*Fundamentals of Computer Aided Geometric Design*, AK Peters Ltd., 1993.

[10] J. Hoschek and H. Pottmann. Interpolation and Approximation with Developable B-Spline Surfaces. In *Mathematical methods for Curves and Surfaces*, Eds. M. Dæhlen, T. Lyche and L.L. Schumaker, Vanderbilt University Press, 255-264, 1995.

[11] J. Hoschek and M. Schneider. Interpolation and Approximation with Developable Surfaces. In *Curves and Surfaces with Applications in CAGD*, Eds. A. Le Méhauté, C. Rabut and L.L. Schumaker, Vanderbilt University Press, 185-203, 1997.

[12] M.J. Mancewicz and W.H. Frey. Developable surfaces properties, representations and methods of design. General Motors R & D Publication, 7637, 1992.

[13] H. Pottmann and G. Farin. Developable rational Bézier and B-Spline surfaces. *Computer Aided Geometric Design*, 12, 513-531, 1995.

[14] H. Pottmann, M. Peternell and B. Ravani. Approximation in Line Space. Manuscript, 1998.

[15] B. Ravani and J.W. Wang. Computer aided design of line constructs. *ASME J. Mechanical Design*, 113, 363-371, 1991.

[16] M. Yang and E. Lee. NC verification for wire-EDM using a R-map. *Computer Aided Design*, 28, 733–740, 1996.