THE ANNALS "DUNĂREA DE JOS" OF GALAȚI
FASCICLE V, TECHOLOLOGIES IN MACHINE BUILDING, ISSN 1221-4566
2010

# Multidimensional Fields Modelling Based on B-Spline Technique with Application in Manufacturing Errors Monitoring

Ion Lixandru, Alexandru Epureanu,
Gabriel Frumuşanu, Marian Viorel Crăciun
"Dunărea de Jos" University of Galați

## ABSTRACT

*In this paper we present methods for the assessment of B-spline functions for a given point, by means of tensor fields and methods for the selection of the initial data. On this basis, we developed different algorithms for testing the theoretical solutions we obtained. The assessment of a multivariable B-spline function is made by a two-steps algorithm, in each step being evaluated a B-spline function of one variable. In practice, the components of the vector or matrix of the control points are experimentally determined and, thereforetheir real value is not known except in a rough approximation. Considering the B-spline function depending on these values by minimizing operators as:* $\int[f(t)]^2dt$, $\iint(x(u,v))^2dudv$, $\iiint(x(u,v,w))^2dudvdw$, *out of the minimum necessary conditions for a multivariable function, we get homogenous linear algebraic systems, out of whose analysis we can obtain useful information on the importance of points which measurements are made on.*

**KEYWORDS:** B-spline function, tensorial fields, assessment algorithms, minimizing functional operators, control points, convex combination.

## 1. Introduction

Let us consider $B=\{e_k\}_{k=\overline{1,n}}$ - the canonical base of the real linear space.

The tensor of order zero is a scalar and the tensor of order one is a vector $u=(u_1,u_2,...,u_n)^t$ and therefore the tensor fields of order zero or one are scalar or vector fields.

The tensor of order two is determined by a matrix $T=(T_{kh})_{\substack{k=\overline{1,m}\\h=\overline{1,n}}}$, where $T_{kh}$ are the components of the tensor T in the orthonormal basis $\{e_k\otimes e_h\}$.

The tensor of order three is given by a 3D matrix (with lines, columns and layers) $T=(T_{ijk})_{\substack{i=\overline{1,m}\\j=\overline{1,n}\\k=\overline{1,p}}}$, where $T_{ijk}$ are the components of the tensor T in the basis $\{e_i\otimes e_j\otimes e_k\}$.

We also note the tensor product with $\otimes$, and the contraction product of index $i$ with $\odot_i$. The last one produces a new tensor whose parts are to be found by summing the components of the two tensors after the common index, $i$. In [5] and [6] the above operations are presented as it follows:

1) Being given

$$u=(u_1,u_2,...,u_m)^t;\ v=(v_1,v_2,...,v_n)^t,\ T=u\otimes v$$

is the tensors composition between u and v. T is a tensor of second order which has the matrix $T=(T_{kh})_{\substack{k=\overline{1,m}\\h=\overline{1,n}}}$, $T_{kh}=u_kv_h, \forall k=\overline{1,m};\forall h=\overline{1,n}$.

2) If $v=(v_1,v_2,...,v_n)^t;\ T=(T_{jk})_{j,k=\overline{1,n}}$, then $U=v\otimes T$ is a tensor of the third order:

$$U=(U_{ijk});U_{ijk}=v_iT_{jk}, \forall i,j,k=\overline{1,n}.$$

3) Being given $T=(T_{ij})_{i,j=\overline{1,n}}$; $U=(U_{kh})_{k,h=\overline{1,n}}$, $W=T\otimes U$ is a tensor of the fourth order;

$$W=(W_{ijkh});W_{ijkh}=T_{ij}U_{kh}, \forall i,j,k,h=\overline{1,n}.$$

4) If $u = (u_1, u_2, ..., u_n)^t$ ; $v = (v_1, v_2, ..., v_n)^t$ , then: $u \odot_i v = v_i u_i = v^t u$ , $u \odot_j v = u_j v_j = u \cdot v$ , that is the usual scalar composition.

5) Given $A = (a_{ij})_{i,j=\overline{1,n}}$ ; $v = (v_1, v_2, ..., v_n)^t$ v, then $A \odot_j v = Av; A \odot_i v = v^t A$ .

6) Given $A = (a_{ij}); B = (b_{kl}); \; i, j, k, l = \overline{1,n}$ , then $T = A \odot_j B = AB = (T_{ik})$, $T_{ik} = a_{ij} b_{jk}$ ; $\forall i, k = \overline{1,n}$ .

$T = A \odot_i B = B^t A = (T_{jk})$,

$T_{jk} = b_{ji} a_{ik}$ , $\forall j, k = \overline{1,n}$ .

Note: $A \odot_i B = B^t \odot_j A$ . Generally, the contraction product of two tensors produces a new tensor whose order equals the sum of the two tensors orders minus two. A B-spline function of one real variable is expressed by means of a set of B-spline basis functions of a certain order which, in their turn, are convex combinations of B-spline basis functions of a lower order. Proceeding further, we come to a B-spline function of order one which is 1 on an interval and 0 for the rest. Starting from here, in [4] we work out an assessing algorithm for the value of a B-spline function of one real variable in a given point. As multivariable B-spline functions are tensor products of B-spline functions of one variable, their assessing algorithm assumes the recurrence of the same calculation algorithm for each coordinate direction. In practice, the components of the vector (or matrix) of the control points are experimentally determined and therefore we do not know their real value, but an approximation of them.

Considering that the B-spline function depends on those values and following the minimizing of some operators as: $\int [f(t)]^2 dt$ , $\int\int (x(u,v))^2 dudv$ , $\int\int\int (x(u,v,w))^2 dudvdw$ etc.

It results linear equations systems out of whose analysis we can conclude on the importance of the points at which measurements are made. Using this in [2] we present the development of an algorithm for the on-line modeling of the thermo-mechanical fields' dynamics of the technological systems. In [1] and [3] we deal with problems connected to computational and algorithmic aspects in designing curves and surfaces, useful in computer graphics, some of these being solved with the help of B-spline functions.

Cutting process evolution is always accompanied and influenced by the existence of specific thermal and mechanical fields. For example, the thermal field, generated because the energy used during the process is transformed in heat and modifies the temperature in different structure points. Another example is structure deformation, meaning that each structure point moves in relation to its initial position, generating the displacements field. Among these fields there can be distinguished cause – fields (e.g., forces field), effect – field (e.g., deformations field) and behavioral fields (e.g., machining precision field).

Most of the manufacturing systems thermo-mechanical fields are multidimensional, generally speaking; for example, in every point owning to an accelerations field, we have the following dimensions: direction (meaning 3 parameters, in 3D), amplitude, frequency, phase, to whom the time (dynamics) always had to be addicted. It means that, to model such a field, we need to use appropriate multi-dimensional models.

The rest of the paper is structured as it follows: In Section 2, we formulate the problem of the assessment of a B-spline function in a given point and establish the importance of measurement knots. In Section 3, there are presented theoretical aspects concerning the B-spline functions. In Section 4, an example of application concerning the dynamics of a mechanical field modelling is shown. Last section concludes the paper on.

## 2. Problem Formulation

Thermo-mechanical fields found in the technological systems' practice are defined along a line, a surface or in a three-dimensional space. In addition, the idea of adding the fourth dimension - the time - is permanent. Therefore the need of modelling a multi-dimensional field appears. In order to do that, we have to work out algorithms for the calculation of the coefficients which express the value of a one-dimensional field in a given point depending on the initial control points. By generalization, we obtain two-step algorithms for the approximation of the bi- or multi- dimensional fields in one of their points.

Another problem, accounted for by practical grounds, is the reduction of the number of sensors where the measurements are made. To do this, we have to work out identity algorithms of the more influential (important) points. On the basis of the above mentioned facts, this paper is concerned with:

- The recursive assessment of a B-spline function of one variable, in a given point, on the basis of the values measured in the initial control points (control points vector);
- The identification of the most important control points by minimizing the operator $\int [f(t)]^2 dt$ ;
- The assessment of a multivariable B-spline function in a given point on the basis of the values measured in the initial control points (control points matrix);

- The identification of the most important control points, by minimizing the operators $\int\int (x(u,v))^2 dudv$, $\int\int\int (x(u,v,w))^2 dudvdw$.

## 3. B-spline Functions

The recursive assessment of the one-dimensional B-spline function uses convex combinations, a fact which gives the assessment process a numerical stability. Based on the assessment algorithm of the one-dimensional B-spline function, by generalization, we can get two-step assessment algorithms of the two or more variables B-spline functions. By minimizing some functional operators, we obtain homogenous systems, out of whose analysis we can determine more important (influential) measurement points.

### 3.1. B-Spline Functions of One Variable

The set of „k+1" consecutive intervals, on which a basic B-spline function, of order k is nonzero, is called its support.

A B-spline function of one real variable is a polynomial segmentary function which is expressed using of a set of basic B-spline functions on each interval, related to a set of knots. Basic B-spline functions are chosen so that the number of supporting intervals to be as small as possible and an eventual modification of a point should affect only the neighbouring parts. Basic B-spline functions are obtained by a recursive process, as it follows:

$$N_{i,1}(t) = \begin{cases} 1, & t \in [t_i, t_{i+1}) \\ 0, & \hat{\imath}n\, rest \end{cases} \tag{1}$$

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

$k \geq 2$. By means of basic B-spline functions, a B-spline function of order k can be written as:

$$f(t) = \sum_{i=1}^{n} d_i N_{i,k}(t), \tag{3}$$

on the knot set there results $(t_i)_{i=\overline{1,n+k}}$. If we write:

$$\begin{cases} N = N(t) = \left(N_{1,k}(t), N_{2,k}(t), \ldots N_{n,k}(t)\right)^t \\ d = (d_1, d_2, \ldots, d_n) \end{cases} \tag{4}$$

then (3) can be written as a contraction product:

$$f(t) = d \odot_i N \tag{5}$$

Using the mathematical induction method, the following property can be proved:

$$\sum_{i=1}^{n-k} N_{i,k}(t) = 1, \forall t \in [t_k, t_{n-k})$$

(6)

Relying on this property, we infer that on the interval $[t_k, t_{n-k+1}]$, the functions $N_{i,k}(t)$ are positive and have the sum equal to 1, so they can be the coefficients of a convex combination. This ensures the numerical stability of the evaluation process of a B-spline function at a point $t \in [t_i, t_{i+1}]$.

To assess the B-spline function $f(t) = \sum_j d_j N_{j,k}(t)$ at a point $t \in [t_i, t_{i+1})$ it is necessary to calculate the nonzero numbers $N_{j,k}(t), j = i-k+1, \ldots, i$.

Using the recurrence relationship (3), we can infer $f(t)$, in relation to the basic B-spline functions of a lower order, as it follows:

$$f(t) = \sum_{j=i-k+1}^{i} d_j \left[\frac{t - t_j}{t_{j+k-1} - t_j} N_{j,k-1}(t) + \frac{t_{j+k} - t}{t_{j+k} - t_{j+1}} N_{j+1,k-1}(t)\right] = \sum_{j=i-k+1}^{i} d_j \frac{t - t_j}{t_{j+k-1} - t_j} N_{j,k-1}(t) +$$

$$+ \sum_{j=i-k+1}^{i} d_j \frac{t_{j+k-1} - t}{t_{j+k-1} - t_j} N_{j,k-1}(t) = \sum_{j=i-k+1}^{i} \left(\frac{t - t_j}{t_{j+k-1} - t_j} d_j + \frac{t_{j+k-1} - t}{t_{j+k-1} - t_j} d_{j-1}\right) N_{j,k-1}(t)$$

If we note:

$$d_j^1(t) = \frac{t - t_j}{t_{j+k-1} - t_j} d_j + \frac{t_{j+k-1} - t}{t_{j+k-1} - t_j} d_{j-1}, \tag{7}$$

we obtain:

$$f(t) = \sum_{j=i-k+1}^{i} d_j^1 N_{j,k-1}(t), \tag{8}$$

or shortly $f(t) = \sum_i d_i^1 N_{i,k-1}(t)$.

Continuing the reasoning and recording:

$$d_i^j(t) = \begin{cases} d_i, & j = 0 \\ \dfrac{t - t_i}{t_{i+k-j} - t_i} d_i^{j-1}(t) + \dfrac{t_{i+k-j} - t}{t_{i+k-j} - t_i} d_{i-1}^{j-1}(t), & j > 0 \end{cases} \tag{9}$$

or

$$d_i^j = \lambda d_i^{j-1} + (1 - \lambda) d_{i-1}^{j-1} \tag{10}$$

where $\lambda = \dfrac{t - t_i}{t_{i+k-j} - t_i}$, we get:

$$f(t) = \sum_i d_i^j(t) N_{i,k-j}(t) \tag{11}$$

But we have

$$N_{i,1}(t)=\begin{cases}1, t\in[t_i,t_{i+1})\\ 0, in\, rest\end{cases}\Rightarrow f(t)=d_i^{k-1}(t), \forall t\in[t_i,t_{i+1})$$

and thus $f(t)$ can be inferred from $d_{i-k+1},...,d_i$ relying on convex combinations (10).

Consequently, first we have to find the index i for which $t\in[t_i,t_{i+1})$, and then we create a table for the recursive assessment of the B-spline functions.

**Table 1** *Recursive assessment of B-spline functions*

| | | | | |
|---|---|---|---|---|
| $d^0_{i-k+1}(t)$ | | | | |
| $d^0_{i-k+2}(t)$ | $d^1_{i-k+2}(t)$ | | | |
| ... | ... | ... | ... | ... |
| $d^0_{i-1}(t)$ | $d^1_{i-1}(t)$ | ... | $d^{k-2}_{i-1}(t)$ | |
| $d^0_i(t)$ | $d^1_i(t)$ | ... | $d^{k-2}_i(t)$ | $d^{k-1}_i(t)$ |

Each entry in the table (except the first column which contains the initial data) is a convex combination of the two adjacent elements in the preceding column and the last term on the right bottom side $\left(d_i^{k-1}(t)\right)$ is the value of $f(t)$ in point $t\in[t_i,t_{i+1})$.

This algorithm can be used to compote the vector of coefficients expressing the value of a B-spline function $f(t)$ in terms of the original control points. We seek coefficients $(\alpha_i)_{i=1}^n$ such that

$$f(t^*)=\sum_{i=1}^n \alpha_i d_i .$$

| ***Algorithm 1***: *Calculus of the vector* $(\alpha_i)_{i=\overline{1,n}}$ *so* $f(t^*)=\sum_{i=1}^n \alpha_i d_i$ *for* $t_i\le t^*\le t_{i+1}$ |
|---|
| 1. create the stocking matrix *mat (k,k)* and the vectors *dp (k), dm (k),* create the vector *res* for the stocking of values $\alpha_i$<br>2. find the index *ind* so $t_{ind}\le t^*\le t_{ind+1}$<br>3. initialize *mat [0] [0] = 1.0*<br>4. *for j = 0* to *k – 2*<br>4.1 $dp[j]=t[ind+j+1]-t^*$<br>4.2 $dm[j]=t^*-t[ind-j]$<br>4.3 for *i = 0* to *j*<br>4.3.1 $temp=mat[i][j]/(dp[i]+dm[j-i])$<br>4.3.2 $mat[i][j+1]+=dp[i]*temp$<br>4.3.3 $mat[i+1][j+1]=dm[j-i]*temp$<br>5. for *i = 0* to *k – 1*<br>5.1 $res[i+ind-k+1]=mat[i][k-1]$ |

In practice, the values of the parts of the control points vector $d=(d_1,d_2,...,d_n)$, are determined experimentally and, therefore, we do not know their real value but their approximation. Considering that the B-spline function depends on these values, we follow the minimizing of the $\int[f(t)]^2dt$ operator. We set the extreme necessary condition $\frac{\partial}{\partial d}\int(f(t))^2dt=0$. We denote:

$$A=\int\left(N\otimes N^t\right)dt=\left(\int N_i(t)N_j(t)dt\right)_{i,j=\overline{1,n}} \quad (12)$$

where: A – mass matrix.

We are thus led to a homogeneous system of n equations with *n* unknown elements for the vector of the control points *d*, of the form:

$$Ad=0 \quad (13)$$

If *rank A = r* (for example, $\Delta_r=det(a_{ij})_{i,j=\overline{1,r}}\neq 0$), then $d_1,d_2,...,d_r$ are main unknowns which are expressed depending on the side unknowns $d_{r+1},d_{r+2},...,d_n$. From a practical point of view, this means that $t_{r+1},t_{r+2},...,t_n$ knots have a greater importance (are more influential) and, therefore, $t_1,t_2,...,t_r$ can be eliminated.

This is illustrated in the following algorithms:

| ***Algorithm** 2: Computation of $N_{i,k}(t)$, given i, k and vec=$(t_i)_{i=1..n+k}$* |
|---|
| function *N(i, k, t, vec)*<br>1. create *res* for storage of return value<br>2. if *k* = = 1<br>2.1 if *t* ≥ *vec(i)* şi *t* < *vec(i)*<br>2.1.1 *res* = 1<br>2.2 else<br>2.2.1 *res* = 0<br>3. else<br>3.1 *res = ( t – vec(i) ) / ( vec(i+k-1) – vec(i) ) * N(i, k-1, t, vec) +*<br>*( vec(i+k) – t ) / (vec (i+k) – vec(i+1) ) * N(i+1, k-1, t, vec)* |

| ***Algorithm 3**: Computation of matrix A elements and of its rank* |
|---|
| 1. for *i = 1* to *n*<br>1.1 for *j = 1* to *n*<br>1.1.2 compute $a_{i,j}$ = *intergral(* $N_{i,k}(t)\cdot N_{j,k}(t)$*, vec(1), vec(n))*<br>using *Algorithm* 1 to compute the values of $N_{i,k}(t)$<br>2. compute the rank of matrix *A* |

### 3.2. B-Spline Functions of Two Variables

Being two B-spline functions of one real variable: $f(u)=\sum_{i=1}^{p} d_i^1 N_{i,k}(u) = d_1 \odot_i N_u$ on the set of nodes $(u_i)_{i=\overline{1,p+k}}$, $g(v)=\sum_{j=1}^{q} d_j^2 N_{j,l}(v) = d_2 \odot_j N_v$ on the set of nodes $(v_j)_{j=\overline{1,q+l}}$, where:

$$N_u = \left(N_{1,k}(u), N_{2,k}(u), \ldots, N_{p,k}(u)\right)^t;$$

$$N_v = \left(N_{1,l}(v), N_{2,l}(v), \ldots, N_{q,l}(v)\right)^t \qquad (14)$$

are the basic B-spline functions corresponding to the functions f and g.

A B-spline function of two variables is defined as the tensor product of two B-spline functions of one-variable:

$$x(u,v) = f(u) \otimes g(v) = d_1 \odot_i \left(N_u \otimes N_v^t\right) \odot_j d_2 . \qquad (15)$$

B-spline basis functions for (15) are the composition of tensors of B-spline basic functions for f and g. Allowing control points to take arbitrary values too, the general formula of a B-spline function of two variables is:

$$x(u,v) = N_u^t d N_v = d \odot_i N_u \odot_j N_v , \qquad (16)$$

where

$$d = \left(d_{ij}\right)_{\substack{i=\overline{1,p}\\ j=\overline{1,q}}} \qquad (17)$$

is the matrix of the control points.

Starting from here, the assessment of a B-spline function of two variables, given a point $\left(u^*, v^*\right)$ is made through a two-steps algorithm:

<u>Step 1</u>. We apply the assessment algorithm of B-spline function of one real variable to each of the columns in the control points matrix $\left(d_{ij}\right)_{\substack{i=\overline{1,p}\\ j=\overline{1,q}}}$ using the knot set: $(u_i)_{i=\overline{1,p+k}}$ and the *u* assessment point $u^*$. The result will be *q* points, one for each column.

<u>Step 2</u>. We apply the same assessment algorithm given the *q* points resulted in step 1, using the knot set: $(v_j)_{j=\overline{1,q+l}}$ and the assessment point $v^*$. The resulting point is the value of $x(u,v)$ in $\left(u^*, v^*\right)$.

On this basis an algorithm can be developed to evaluate a B-spline function of two variables at a given point.

| ***Algorithm 4**: Computation of* $x\left(u^*, v^*\right)$ |
|---|
| 1. for j = 1 to q<br>1.1 using algorithm 1 to compute the coefficients $\alpha_{ij}$ so $x\left(u^*, v_j\right)=\sum_{i=1}^{p}\alpha_{ij}d_{ij}$ , for $u_i \le u^* \le u_{i+1}$<br>1.2 using algorithm 1 to compute the coefficients $\beta_j$ so $x\left(u^*, v^*\right)=\sum_{j=1}^{q}\beta_j x\left(u^*, v_j\right)$, for $v_j \le v^* \le v_{j+1}$ |

Also, in the case of two-variables B-spline functions, the elements of the control points' matrix $d = (d_{ij})_{\substack{i=\overline{1,p}\\ j=\overline{1,q}}}$ could be considered variables, trying to minimize the $\iint (x(u,v))^2\, du dv$ operator. As $x(u,v) = N_u^t d N_v$ we further obtain the equation:

$$\int (N_u \otimes N_u^t) du \odot_j d \odot_j \int (N_v \otimes N_v^t) dv = 0 .$$

If $A_u = \int (N_u \otimes N_u^t) du = \left(\int N_i(u) N_j(u) du\right)_{i,j=\overline{1,p}}$ and $A_v = \int (N_v \otimes N_v^t) dv = \left(\int N_i(v) N_j(v) dv\right)_{i,j=\overline{1,q}}$ we obtain, for the control points' matrix $d$, the system:

$$A_u d A_v = 0 \tag{18}$$

| *Algorithm 5*: *Computation of matrices* $A_u = (a_{i,j})_{i,j=\overline{1,p}}$, $A_v = (b_{i,j})_{i,j=\overline{1,q}}$ *elements and of the matrix of the system* $A_u \cdot d \cdot A_v = O$, rank, given $u=(u_i)_{i=1..p+k}$, $v=(v_j)_{j=1..q+l}$ |
|---|
| 1. for $i = 1$ to $p$<br>1.1 for $j = 1$ to $p$<br>1.1.2 compute $a_{i,j}$ = *intergral*( $N_{i,k}(u) * N_{j,k}(u)$ , *u(1), u(p)*)<br>using *Algorithm* 1 to compute the values of $N_{i,k}(u)$ and $N_{j,k}(u)$<br>2. for $i = 1$ to $q$<br>2.1 for $j = 1$ to $q$<br>2.1.2 compute $b_{i,j}$ = *intergral*( $N_{i,j}(v) * N_{j,l}(v)$ , *v(1), v(q)*)<br>using *Algorithm* 1 to compute the values of $N_{i,k}(t)$ and $N_{j,l}(v)$<br>3. compute the rank of the matrix of the system $A_u \cdot d \cdot A_v = O$ |

The analysis of system (18) can lead to conclusions about the influential points, in which the measurements should be taken, and can be used to develop an algorithm for their determination.

### 3.3. B-Spline Functions of Three Variables

A B-spline function of three variables can be defined as a tensor product of three B-spline functions of one variable or as the tensor product of a B-spline function of two variables and a B-spline function of one variable. If *f, g, h* are three B-spline functions of one variable $f(u) = \sum_{i=1}^{p} d_i^1 N_{i,l}(u) = d_1 \odot_i N_u$ defined on the knot sets $(u_i)_{i=\overline{1,p+l}}$, $g(v) = \sum_{j=1}^{q} d_j^2 N_{j,m}(v) = d_2 \odot_j N_v$ defined on the knot sets $(v_j)_{j=\overline{1,q+m}}$ and $h(w) = \sum_{k=1}^{r} d_k^3 N_{k,n}(w) = d_3 \odot_k N_w$ defined on the knot sets $(w_k)_{k=\overline{1,r+n}}$ where:

$$\begin{aligned} N_u &= (N_{1,l}(u), N_{2,l}(u), \ldots, N_{p,l}(u))^t ; \\ N_v &= (N_{1,m}(v), N_{2,m}(v), \ldots, N_{q,m}(v))^t ; \\ N_w &= (N_{1,n}(w), N_{2,n}(w), \ldots, N_{r,n}(w))^t \end{aligned} \tag{19}$$

are the basic B-spline functions corresponding to the functions f, g, h, then

$$x(u,v,w) = d \odot_i N_u \odot_j N_v \odot_k N_w \tag{20}$$

is a B-spline function of three variables which has as its basic functions the tensorial composition of the basic functions of f, g, h and

$$d = (d_{ijk})_{\substack{i=\overline{1,p}\\ j=\overline{1,q}\\ k=\overline{1,r}}} \tag{21}$$

represents the control points matrix.

Similarly, given the B-spline function of two variables, we can also work out a two steps algorithm for the assessment of the function $x(u,v,w)$ in a given point $(u^*, v^*, w^*)$.

Step 1. We apply the assessment algorithm of a two-variables B-spline function to each points layer of the control points matrix using the knot set $(u_i)_{i=\overline{1,p+l}}$; $(v_j)_{j=\overline{1,q+m}}$ and the assessment point $(u^*, v^*)$. The result will be $r$ points, one for each layer.

Step 2. We apply the assessment algorithm of a B-spline function of one real variable to the $r$ points resulted in step 1, using the knot set $(w_k)_{k=\overline{1,r+n}}$ and the assessment point $w^*$. Resulting point represents the value $x(u^*, v^*, w^*)$.

This allows the implementation of an algorithm for evaluating a B-spline function with three variables at a given point.

| ***Algorithm 6:*** *Calculation of the value* $x(u^*, v^*, w^*)$ |
|---|
| 1. for k = *1* to *r* |
| 1.1 use *Algorithm 2* to calculate coefficients $\alpha_{jk}$, so that $x(u^*, v^*, w_k) = \sum_{j=1}^{q} \alpha_{jk} x(u^*, v_j, w_k)$, for $u_i \le u^* \le u_{i+1}$, $v_j \le v^* \le v_{j+1}$ |
| 2. use *Algorithm 1* to calculate coefficients $\beta_k$, so that $x(u^*, v^*, w^*) = \sum_{k=1}^{r} \beta_k x(u^*, v^*, w_k)$, for $w_k \le w^* \le w_{k+1}$ |

Similarly the B-spline function of two variables, trying to minimize the $\iiint (x(u,v,w))^2 dudvdw$ operator, having the elements of the control points matrix $d = (d_{ijk})_{\substack{i=\overline{1,p} \\ j=\overline{1,q} \\ k=\overline{1,r}}}$ as unknowns, we obtain:

$$\frac{\partial}{\partial d} \iiint (x(u,v,w))^2 dudvdw =$$

$$= \frac{\partial}{\partial d} \iiint \left(N_u^t d N_v M_w\right)^2 dudvdw = 0$$

In tensorial notation this is written:

$$\iiint \frac{\partial}{\partial d} (d \odot_i N_u \odot_j N_v \odot_k N_w)^2 dudvdw =$$

$$= 2 \iiint (d \odot_i N_u \odot_j N_v \odot_k N_w) \otimes$$

$$\otimes \left(N_u \otimes N_v^t \otimes N_w^t\right) dudvdw .$$

In this way we get the equation:

$$A_u \odot_i d \odot_j A_v \odot_k A_w = 0 \qquad (22)$$

and relying on it we can conclude the importance of the measurement control points.

This can be put into practice through the following algorithm.

| ***Algorithm 7****: Computation of matrices* $A_u = (a_{i,j})_{i,j=\overline{1,p}}$, $A_v = (b_{i,j})_{i,j=\overline{1,q}}$, $A_w = (c_{i,j})_{i,j=\overline{1,r}}$ *elements and of the matrix of the system* $A_u \otimes_i d \otimes_j A_v \otimes_k A_w = O$ *rank, given* $u=(u_i)_{i=1..p+k}$, $v=(v_j)_{j=1..q+l}$, $w=(w_j)_{j=1..r+k}$ |
|---|
| 1. for *i* = *1* to *p* |
| 1.1 for *j* = *1* to *p* |
| 1.1.2 compute $a_{i,j}$ = *intergral(* $N_{i,k}(u) * N_{j,k}(u)$ *, u(1), u(p))* |
| using *Algorithm* 1 to compute the values of $N_{i,k}(u)$ and $N_{j,k}(u)$ |
| 2. for *i* = *1* to *q* |
| 2.1 for *j* = *1* to *q* |
| 2.1.2 compute $b_{i,j}$ = *intergral(* $N_{i,j}(v) * N_{j,l}(v)$ *, v(1), v(q))* |
| using *Algorithm* 1 to compute the values of $N_{i,k}(t)$ and $N_{j,l}(v)$ |
| 3. for *i* = *1* to *r* |
| 3.1 for *j* = *1* to *r* |
| 3.1.2 compute $c_{i,j}$ = *intergral(* $N_{i,j}(w) * N_{j,l}(w)$ *, w(1), w(r))* |
| using *Algorithm* 1 to compute the values of $N_{i,k}(w)$ and $N_{j,l}(w)$ |
| 4. compute the rank of the matrix of the system $A_u \otimes_i d \otimes_j A_v \otimes_k A_w = O$ |

## 4. Dynamics of the Elastic Return Mechanical Field of a Folded Piece Modelling

The case of a piece of tin, obtained by bending in a die was considered (Fig. 1); the manufacturing method can be applied in two versions: with or without retaining the metal sheet during deformation. Notations of Fig. 1 have the following meanings: 1 - die, 2 - punch, 3 - retention plate, 4 - piece.

It is known that after working through this process, when they are pulled from the die, the pieces undergo elastic recovery; it leads to a difference between the shape and size since the end of processing (fig.1-b, 1) and those after releasing the piece from the die (Fig. 1-b, 2).

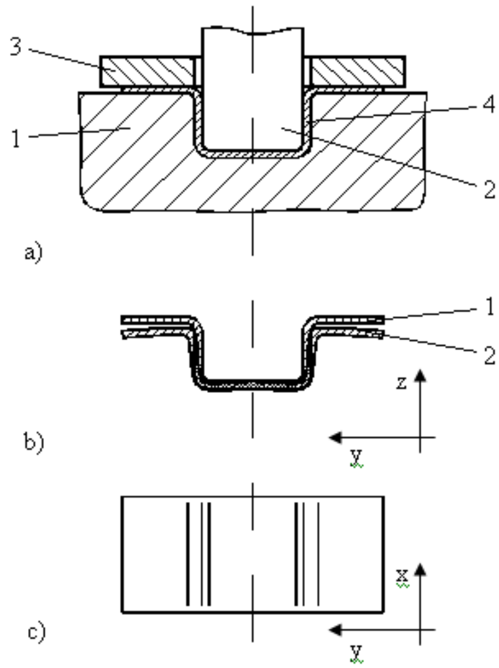


**Fig.1.** *The manufacturing scheme*

**Table 2** *Network points co-ordinates*

| Run | X [mm] | Y [mm] | Z [mm] |
|---|---|---|---|
| 1 | 20.2222805 | 1.01142645 | 2.93657732 |
| 2 | 20.2220116 | 2.03363919 | 2.93871427 |
| 3 | 20.2213116 | 3.05495787 | 2.94191194 |
| 4 | 20.2205658 | 4.07670498 | 2.94423175 |
| 5 | 20.2194958 | 5.09782887 | 2.94783568 |
| 6 | 20.2185631 | 6.11980772 | 2.95016265 |
| 7 | 20.2171822 | 7.14081621 | 2.95249176 |
| 8 | 20.2159843 | 8.16330910 | 2.95401502 |
| 9 | 20.2146492 | 9.18423748 | 2.95697737 |
| 10 | 20.2136822 | 10.2065163 | 2.96184993 |
| … | … | … | … |
| 5419 | 49.9945183 | 79.2478180 | 105.399284 |
| 5420 | 49.9937019 | 80.2482605 | 105.399307 |
| 5421 | 49.9944992 | 81.2486801 | 105.399307 |
| 5422 | 49.9947624 | 82.2.492523 | 105.399269 |
| 5423 | 49.9960632 | 83.2493515 | 105.399292 |
| 5424 | 49.9953613 | 84.2498016 | 105.399300 |
| 5425 | 49.9959068 | 85.2496948 | 105.399338 |

So, we are dealing with a mechanical field of deformations induced by the phenomenon of elastic return, which can be modelled (both actual and as dynamics) by using the new type of spline functions.

On the sheet surface, before bending was traced a network of points whose coordinates were measured at the end of deformation and after removal from the die (after elastic recovery had occurred). For reasons of symmetry, only the left half of the piece was considered. Measured coordinates of points making up the network are presented in tables as Table 1 (this one includes details from the end of the bending without restraint).

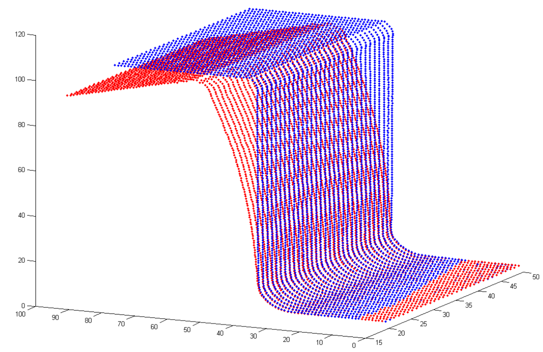

**Fig.2.** *Worked piece before and after elastic recovery*

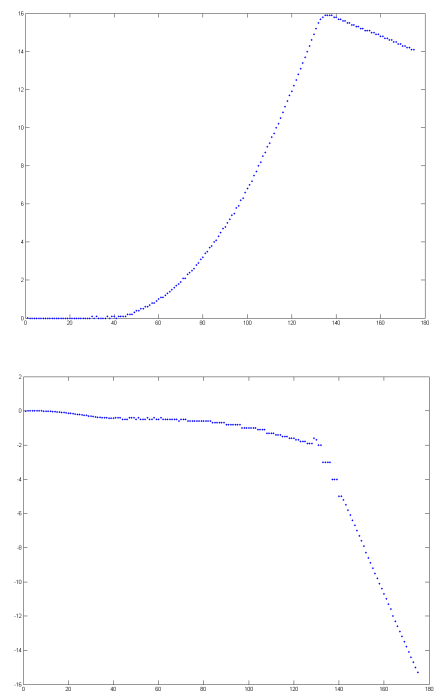

**Fig. 3.** *Model of the difference function, based on the 175 points, when bending without retention* b)
*a – along z axis; b - along y axis*

For modelling the dynamics of the elastic recovery process, by using the new type of spline functions, the following methodology was used:

- we have modelled both the state in the end of the process of bending (as initial reference state) and the state after the elastic return (as final state), based on the points measured coordinates;

- it was determined the difference function Δf, characterizing the transition from the initial state to the final one $\Delta f(x,y,z) = f_i(x,y,z) - f_f(x,y,z)$;
- the difference function was modelled using the new type of spline functions; starting from the observation that points arranged in planes x = ct (175 in each plane) have identical behaviour, the behaviour of the difference function was examined only in such a plane (Fig. 3, 4, for the case of bending without retention); we used successively all the 175 points and, respectively, only 5 among them, chosen as the most significant by using a genetic algorithm (Fig. 5, 6 for bending without/with retention);
- an addition was made between the known values, in the reference state, and those obtained by spline modelling the difference function and then we compared the results to the situation as modelled on the first stage, based on the known values in all 175 points.

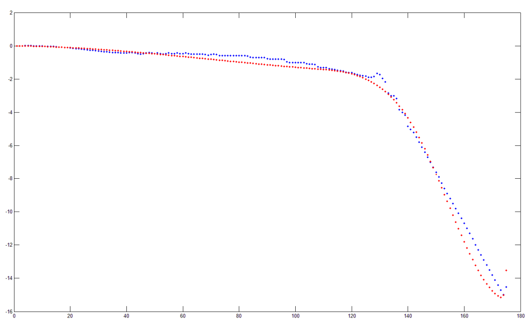

**Fig. 4.** *Comparison between the interpolation into z direction based on 175 / 5 points (blue / red)*

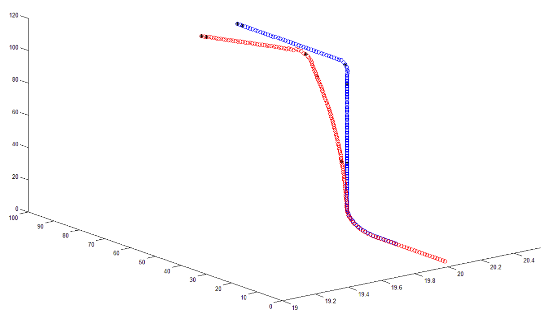

**Fig. 5** *The 5 points selected by the genetic algorithm (* - Black), when bending without restraint*

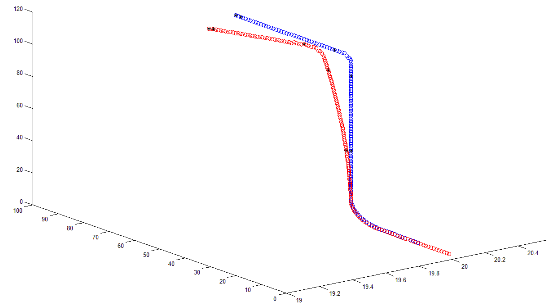

**Fig. 6** *The 5 points selected by the genetic algorithm (* - Black), when bending with restraint*

Since the graphics for the second experiment (bending with retention) are very similar to those made for the first one, we have explicitly presented in this case, in order to make a comparison, only the position of the five representative points chosen by the genetic algorithm.

We calculated the mean square deviation between the measured positions and those obtained by modelling of the 175 points considered and we found the value of 0.9222 (bending without restraint) and 0.4164 (bending with retention).

## 5. Conclusions

In order to model and identify the mechanical fields of different technological systems and their dynamics, we have to work out algorithms based on the coherence of these fields.

To substantially facilitate the reduction of the number of sensors needed to dynamically check the thermo-mechanical fields, we have to find algorithms for the calculation of the more important measurement points, to use them for the adaptive management of the technological systems.

The use of tensors easily permits passing from B-spline functions of one real variable to multivariable B-spline functions. The assessment algorithm of a B-spline function of one real variable, based on certain convex combinations, ensures the numerical stability of the process.

Multivariable functions can be assessed by means of a two steps algorithm, each of the steps assessing a B-spline function of one real variable. The tensor approach of B-spline functions allows obtaining useful practical information about the importance of measurement knots, by minimizing some operators.

**Acknowledgement**

The authors gratefully acknowledge the financial support of the Romanian Ministry of Education, Research Youth and Sports through Grant PN II-IDEAS-653/2007.

## References

**1. Alhanaty, M. , Bercovier, M.,** *Curve and Surface Fitting and Design by Optimal Control Methods*. CAGD, (33): 167-182, 2001.

**2. Lixandru, I., Epureanu, Al., Frumusanu, G., Crăciun, M.V.,** *Algorithm Developing for Online Modelling of Technological Systems' Thermo-mechanical Fields' Dynamics*. Workshop on Mathematics, Computer Science and Applications. November, 7-9, 2008. Conference room: SA (Library) Department of Mathematics and Computer Science. The Annals of “Dunărea de Jos” University of Galați. ISSN 1842 – 6506. Fascicle II, pp126–134, Mathematics, Physics, Chemistry, Informatics. Thematic Series, 2008.

**3. Oproiu, V.,** *Geometria Computațională a Curbelor şi Suprafețelor*. Ed.Universității“Al.I.Cuza” Iaşi, pp. 97-113, 2003.

**4. Sherar, P.A.,** *Variational Based Analysis and Modelling using B-splines*. Cranfield University-School of Engineering. BPhD Thesis, Academic Year 2003-2004, pp.16-42; 50-62.

**5. Soos, E., Teodosiu, C.,** *Calcul Tensorial cu Aplicații în Mecanica Solidelor*, Ed. Ştiințifică şi Enciclopedică, Bucureşti, pp. 38-67, 1981.

**Modelarea câmpurilor multidimensionale pe baza tehnicii b-spline cu aplicații în monitorizarea erorilor de fabricație**

Rezumat

În această lucrare sunt prezentate metode pentru evaluarea funcțiilor B-spline într-un punct dat, prin intermediul câmpurilor de tensori şi metode pentru selectarea datelor inițiale.

Pe aceste baze, am dezvoltat diverşi algoritmi pentru testarea soluțiilor teoretice obținute. Evaluarea unei funcții B-spline de mai multe variabile este realizată printr-un algoritm în doi paşi, în fiecare pas fiind evaluată o funcție B-spline a unei variabile.

În practică, componentele vectorului sau matricii punctelor de control sunt determinate experimental şi de aceea acestea nu sunt cunoscute ca valoare reală ci doar aproximate.

Considerând funcția B-spline dependentă de aceste valori, prin minimizarea operatorilor ca: $\int\left[f(t)\right]^2 dt$, $\iint\left(x(u,v)\right)^2 dudv$, $\iiint\left(x(u,v,w)\right)^2 dudvdw$, conduce la condiția minimă necesară pentru o funcție de mai multe variabile.