Proceedings of the IMProVe 2011
International conference on Innovative Methods in Product Design
June 15th – 17th, 2011, Venice, Italy

# Constrained fitting of B-Spline curves based onthe Force Density Method

L. Barbieri[(a)], F.Bruno[(a)], M.Muzzupappa[(a)], J.-P.Pernot[(b)]

[(a)]University of Calabria, Rende, Italy
[(b)]LSIS, UMR CNRS 6168, Aix-en-Provence, France

## Article Information

Keywords:
*B-Spline curve,*
*Constrained fitting,*
*Force Density Method.*

Corresponding author:
*Loris Barbieri*
*Tel.: +390984494667*
*Fax.: +390984494663*
*e-mail: loris.barbieri@unical.it*
*Address:Arcavacata di Rende (CS) ponte P.Bucci, cubo 45C.*

## Abstract

*This paper presents a novelapproach for constrained B-Spline curve approximation based on the Force Density Method (FDM). This approach aims to define a flexible technique tool for curve fitting, which allows approximating a set of points taking into account shape constraints that may be related to the production process, to the material or to other technological requirements.*
*After a brief introduction on the property of the FDM and the definition of the network usedfor the formulation of the fitting problem, the paper explains in detail the mathematical approach, the methods and the techniques adopted for the definition of the proposed constrained B-Splinecurve approximation.*
*The results suggest that the adoption of a mechanical model of bar networks allows developing a more flexible tool than the traditional least squared methods (LSM) usually adopted for fitting problems. Numerical examples show that the new approach is effective in fitting problems when the satisfaction of shape constraints, such as those related to production orto technological processes, are required.*

## 1 Introduction

Engineers often need to fit geometric or experimental data using a model based on mathematical equations. Several mathematical models are commonly used for curve fitting (e.g.: polynomial, Bézier, B-Spline, NURBS) and the choice of the right model requires a good knowledge of both the data to be fitted and the formulation that may be adopted. But, once the model has been chosen and the data are approximated through one or more curves, engineers can determine important characteristics of the data, such as the rate of change along the curve (first derivative), the local minimum and maximum points of the function (zeros of the first derivative), and the area under the curve (integral).

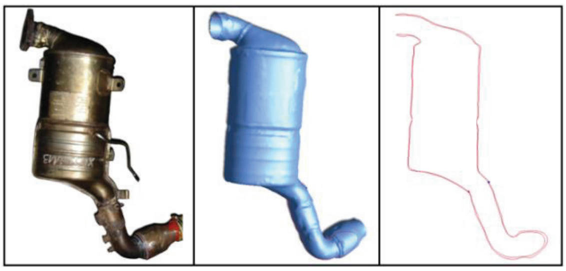
The goal of curve fitting is to find the set of parameters that most closely match the data. The main theory behind curve fitting data revolves around minimizing the sum of the squares of the residuals: the residual of a curve fit for each data point typically is the vertical deviation from the observed data point and the predicted value as given by the function of the curve (the Euclidean distance, rather than vertical distance, might have been used, but it is more difficult to handle because it produces non-linear equations [1]). This approach is known as the least squares method (LSM). This method assumes that the measurement errors are independent and normally distributed with constant standard deviation. For linear functions, the solution for a best fit curve is a defined closed solution that can be directly solved.

For non-linear functions, an iterative non-linear least squares approach is utilized to converge to the best fit curve (several algorithms have been formulated to aid in converging the solution to non-linear curve fitting [2-4]).

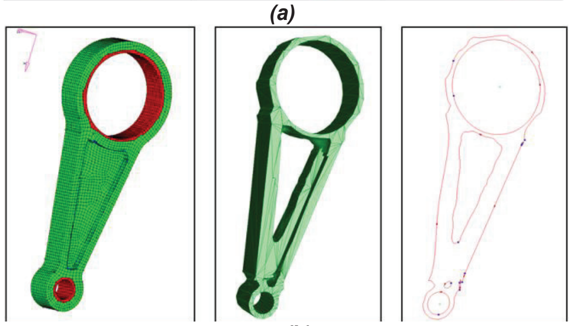
Starting from a set of points, that could be captured as part of a reverse engineering process or extracted from topological optimization results, the designer has the necessity to rework these raw data with the introduction of

LSM is the most widespread approach for curve fitting but, such as for other optimization methods [5-8], in some technological fields it is not easily applicable.

Reverse Engineering and Topological Optimization are two application fields in which LSMs are often used to approximate a set of points or polygons with a mathematical model (fig.1).



***(a)***



***(b)***

***Fig.1Physical object, mesh model and section extracted from the mesh model(a). Voxel model, polyhedron model and its silhouette(b).***

shape constraints in order to produce B-Spline curve that successfully reproduce regularities required by engineering applications (such as curvature, position and distance constraints). The constraints to be satisfied may be related to the production process, to the material or to other technological issues.

Only the specialist that holds the specific knowledge is able to interpreter these data and take the right decisions about the fitting (e.g.: mathematical model and relative parameters, how to split the curve in several segments, continuity between each couple of consecutive segments, etc.).

So it is possible to assert that curve fitting in some cases is a knowledge-based process, whichmight not be completely automated.

To better control the link between the semantic of the processes, to avoid low manipulations of geometric data, and to be more flexible thanthe LSMs, the Force Density Method (FDM) introduced by Scheck [9] has already proved its capacities. It is based on a mechanical model of a network with tensile bars.Thanks to its properties, the FDM has been applied in various domains including lightweight tension and tensile integrity structures [10], optimization of tensile textile structures [11], and deformation of free form surfaces [12,13]. The large spread of this method is due to its main advantages:

- Any state of equilibrium of a general net structure can be obtained by the solution of one system of linear equations. This system is constructed using the force-length ratios or “force densities” in the branches as network description parameters. In other words, one single force density is prescribed in order to obtain a unique result for the appropriate state of equilibrium.
- It is possible to extend the linear approach to the non-linear force density method. The number of non-linear equations is identical with the number of additional conditions and is independent of the number of net nodes [9].

Therefore this approach may be simpler and less expensive than the conventional solutions.

In this paper, we propose to take advantage of the FDM to develop a constrained B-Splinefitting technique,able to take into account some shape constraintsthat may be specified in order to satisfy specific technological needs.

The paper describes the overall methodology necessary to reach a constrained fitting of B-Spline curves based on the FDM in which only a few set of constrains have been tested. After a brief introduction on the property of the FDM and the definition of the network adopted for the formulation of the fitting problem, the paper explains in detail the mathematical approach,the methods and the techniques usedfor the definition of the proposed constrained B-SplineSplinecurve approximation.

The approach is composed by four stages. In the first stage an initial B-SplineSplinecurve fitting problem is solved using the minimization of an objective function defined as the sum of the square external forces computed on the free nodes of the network.

The second stage implies the use of a toolbox for the curvature analysis and evaluation. In addition to the curvature ,this toolbox is able to compute and depict the most important points belonging to the B-SplineSpline by means of the Leyton Process-Grammar [14].

The third stage is characterized by a splitting toolbox that, using the information elaborated in the previous stage, introduces $G^0$ continuities on the initial curve in order to simplify its manipulation. In fact, in the last stage, the B-Spline fitted curve is calculated again with the introduction of some shape constraints like position, distance, straight and curvature.The fitting problem is solved using the Optimization Toolbox of MatLab[15] that allows to program linear/non-linear and equality/inequality constraints.

The results suggest that the adoption of a mechanical model of bar networks allows developing a more flexible tool than the traditional LSMs usually adopted for fitting problems. Numerical examples show that the new approach is effective in fitting problems when the satisfaction of shape constraints, such as those related to production or to technological processes, are required.

## 2 State-of-the-art

Even if the FDM has been applied in different and various engineering fields, for what concern curve fitting problems the state-of-the-art proposes only two researches.

Eric Saux[16] proposes an application to cartographic generalization of maritime lines. His attempt is to include the fitting method for geographic data reduction in cartographic maps; in particular the use of B-SplineSplinecurves for modeling smooth lines such as roads, railways or waterways. With his research Saux proposes two approaches for the implementation of the working tools in generalization of cartographic maps[17]. One is automatic while the other is interactive. A catalogue of cartographic generalization operators has been proposed, including selection/elimination, aggregation, structuring, compression (or filtering), smoothing, exaggeration, caricaturing, enlargement and displacement. The curve deformation is obtained through mechanical parameter modifications which lead to a shape modification, the strategy is that suggested by Léon and Trompette[18] based on the ForceDensity Method. Each equilibrium position of the barsnetwork (coupledto the control polygon of the curve) can be determined solving a linear system of equations. The strategy relies on a mechanical approach permitting fast calculations as well as local and global deformations.

Sanchez [19] proposes NURBS fitting techniques to represent tensile structures. The first step of the method consists into an initial grid generation; this structural net has the minimum nodes and elements needed to obtain the double curvature of the membrane. On the initial grid the FDM is applied in order to obtain the equilibrium shape of the membrane. Once the equilibrium shape is obtained, NURBS fitting techniques [20, 21] are used to represent a smooth parametric surface that passes through the points obtained in the previous form finding step. The values of the force density parameters and the applied forces may also be changed, obtaining different equilibrium shapes in real time. With this method they have tested different algorithms, but these are not described in detail in the paper.

The maindrawback of these researches is their being limited to the contexts of cartographic map and tensile structures,respectively. Furthermore they don’t consider the possibility to enforce shape constraints that may be required by engineering applications and technological processes.

## 3 Proposed method

The proposed approach for performing a constrained Force Density B-SplineSplinecurve fitting is based on a four stages method depicted infig. 2.

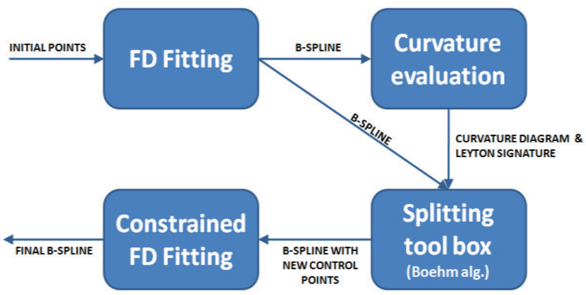



***Fig.2Stages for the constrained FD fitting method.***

In the first stage we present a B-Splinecurve fitting procedure in which a fixed number of control points are the only unknowns, and they are solved by minimizing an objective function computed using the FDM.TheB-Spline curve is the model adopted for the FD fitting because it is suitable for CAD/CAM and machining applications. The B-Spline curve $\boldsymbol{B}(u)$of degree $p$ and $n$ control points($\boldsymbol{P}_0 \dots \boldsymbol{P}_{n-1}$)is defined by:

$$\boldsymbol{B}(u) = \sum_{i=0}^{n-1} N_{i,p}(u) \cdot \boldsymbol{P}_i \quad ; \quad u_{min} \leq u < u_{max} \tag{1}$$

where$N_{i,p}(u)$are the basis functions defined by the Cox-de Boor recursion formulas on $[u_i, u_{i+p+1}]$; specifically:

$$N_{i,1}(u) = \begin{cases} 1 \; if \; x_i \leq u < x_{i+1} \\ 0 \; otherwise \end{cases} \tag{2}$$

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

The second stage implies the use of a toolbox for the curvature analysis and evaluation. This toolbox calculates the curvature of the approximating curve and depicts this information on diagrams that allows the user to identify and select, in a more objective manner, the characteristic points of the initial fitted curve. In addition to the curvature graphs, the curvature evaluation toolbox is able to highlight the most important points belonging to the B-Spline by means of the Leyton Process-Grammar.

This information represents the input for the next stage in which a splitting toolbox, whose implementation is based on the Boehm's knot insertion algorithm, allows to increase the flexibility of the B-Spline without changing the shape of the curve by inserting additional knot values into the knot vector and thus inserting new control points. The result is a more flexible curve on which $G^0$ continuities have been introduced.

In the last stage, the B-Spline fitted curve is calculated again with the introduction of some shape constraints like position, distance, straight and curvature.

All the functions have been implemented in MatLab and the solver used for the resolution of the optimization problem is *fmincon* of the Optimization Toolbox [22] that attempts to find a constrained minimum of a scalar function of several variablesby starting at an initial estimate.In the followings sections each stage will be discussed in details.

## 3.1 Initial Fitting

This section describes in detail the mathematical procedure defined for the initial unconstrained fitting.The B-Spline curve is the model adopted for the fitting of $m$ initial $\boldsymbol{Q}_k$ points, the curve's degree $p$ and the number of the $n$ control points $\boldsymbol{P}_i$ of the fitted curve must be input also. The coordinates of the control points of the fitted B-Spline curve are the solutions of a minimization problem in which the objective function is computed by means of a mechanical model of spring network. The following flow chart depicts the mathematical approach defined to reach this goal.

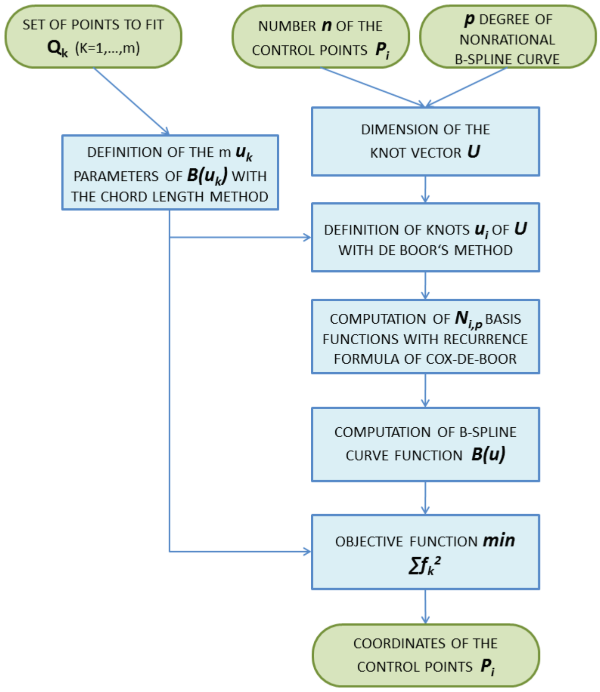



***Fig.3 Flow chart and methods for the FD B-Spline fitting curve.***

For each step, specific functions have been programmed in MatLab, in order to automate the whole process.

From the initial $\boldsymbol{Q}_k$points it is possible to derive the $m$values of the $\bar{\boldsymbol{u}}_k$ parametric coordinates necessary to find the positions of the free nodes usedto define the mechanical model. The Chord Length Method [21] has been adopted in order to computethe $\bar{\boldsymbol{u}}_k$ values. This is the most widely used method, and it is generally adequate.

The $\bar{\boldsymbol{u}}_k$ parameters, in conjunction with the degree $p$ and the number $n$ of control points, are necessary for the computation of the knots vector $\boldsymbol{U}$. The B-Spline curves adopted for the initial unconstrained fitting problem are open uniform B-Spline curves that impose that the initial and final $p+1$ knot values are, respectively, 0 and 1. In presence of internal knots, which number is equal to $(n-1)-p$, their definition is obtained through the adoption of either the Averaging Method or the De Boor Method, that allows knots vector $\boldsymbol{U}$ to reflect the distribution of the$\bar{\boldsymbol{u}}_k$ parameters[21, 23]. Once the knots vector $\boldsymbol{U}$ has been defined, the $N_{i,p}(u)$basis functions and the equation of the B-Spline$\boldsymbol{B}(u)$ curve are computed automatically, in which the unknowns are the $(x_i, y_i)$ coordinates of the control points $\boldsymbol{P}_i$. In order to select one solution among all the possible ones, an *objective function* is added and a criterion has to be chosen.

### 3.1.1 Mechanical model definition

The FDM enables geometric manipulations of a bar network through the modification of external forces applied to its nodes. It is possible to distinguish the blocked nodes from the free nodes (unknowns of the fitting problem). The position of the free nodes is automatically updated to compensate the external forces variations and maintain the static equilibrium of the structure.

Each bar can be seen as a spring of length $l_i$ with a force density (stiffness) $q_i$. To preserve the static equilibrium state of bars, $f_i$external forces have to be applied to the endpoints of the bar: $f_i = q_i \cdot l_i$ (fig.4). The set of external forces applied to the initial bar network can be then obtained through the static equilibrium equations at each free node.

The shape of a network strongly depends on the connectivityrulesspecifyingwhich nodes have to be connected by springs. For the fitting problem the definition of a mechanical model of a spring network has been necessary for the computation of the objective function. Among the different solution that could be adopted the next figure shows the final network that is coupled between the point to be fitted and the B-Spline curve:

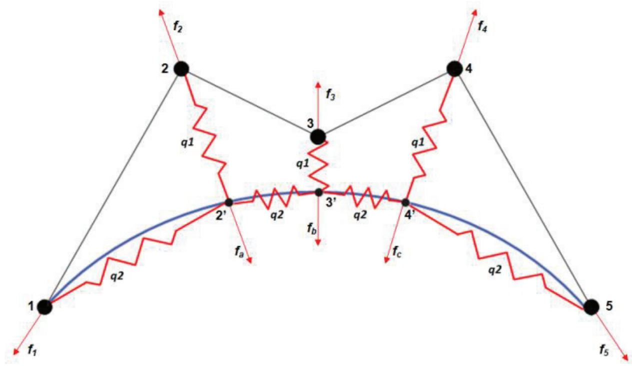



***Fig.4 Mechanical model of a spring network adopted for the fitting.***

The points*1 to 5* are the$\boldsymbol{Q}_k$blocked nodes ($m$ initial points) to fit; *2'*, *3'* and *4'* are the free nodes located directly on the curve. There are totally $2m-3$ springs that can be distinguished in those that link two consecutive nodes on the curve with the same force density $q_2$, and the other ones, with force density $q_{1,}$ that link a blocked node with the correspondent free node on the curve. The force densities $q_i$ are parameter that increase the flexibility of the method indeed their modification influence the shape of the fitted curve. In particular for $q_1 = 1$ and $q_2 = 0$ the method returns the same results as forthe LSM.

As stated before, the equilibrium is calculated only on the free nodes, then, for a fitting problem with $m$ points to fit, $2 \times (m-2)$ equations will be calculated in order to compose the equation of the objective function. The minimization of the sum of the square external forces, computed on the free nodes of the network, has been adopted as objective function.

$$\Phi(F) = \sum_{j=1}^{m-2} f_j^2 = \sum_{j=1}^{m-2} f_{xj}^2 + \sum_{j=1}^{m-2} f_{yj}^2 \tag{4}$$

The equation of the objective function *Φ(F)* is calculated automatically and given as input to the solver, which returns a vector of dimension $2n$ that are the $(x_i, y_i)$ coordinates of the $n$ control points$\boldsymbol{P}_i$.

## 3.2 Curvature Evaluation

The second stage of the proposed methodology is the curvature evaluation, whose aim is the individuation, in an objective manner, of the principal characteristics of the B-Spline fitted curve computed in the previous stage. The curvature evaluation consists of two main steps: the curvature analysis and the Leyton signature that allows to index each curve through the adoption of the Leyton'sprocess-grammar.

### 3.2.1 Curvature Analysis

The curvature is defined as the vector $k$ whose magnitude is the reciprocal of the radius of the circle [24, 25]. For aB-Spline curve ***B****(u)* the curvature function is:

$$k(u) = \frac{\det\left(\frac{dB}{du}, \frac{d^2B}{du^2}\right)}{\left\|\frac{dB}{du}\right\|^3} \tag{5}$$

For example, the next figures depict a cubic B-Spline curve (red) with 14 control points (blue circles) calculated by the fitting of 21 initial points (black circles) and its curvature graph that allows to pick out the points of the curve with the highest curvature.

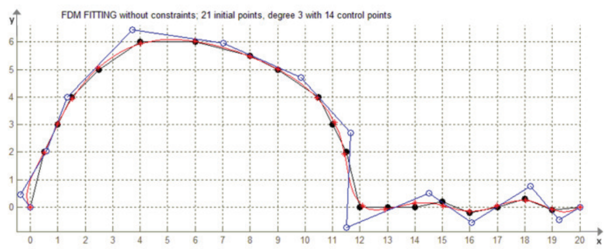



***Fig.5 Initial B-Spline fitted curve.***

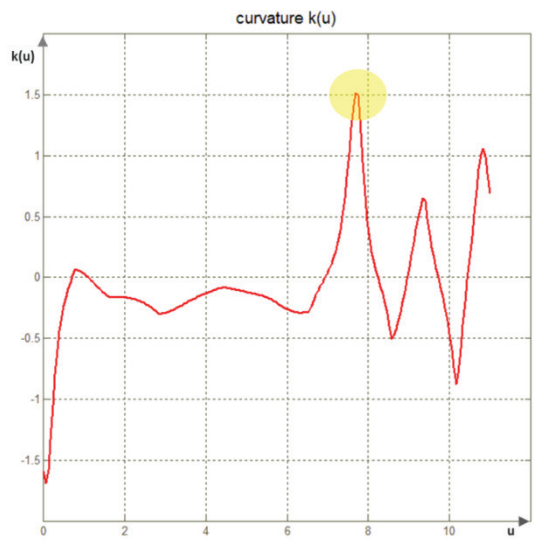



***Fig.6Curvature plot of the initial fitted B-Spline curve.***

### 3.2.2 Leyton's Signature

The process-grammar of Leyton [14, 26] is a syntactic description of a curve based on two structural factors: symmetry and curvature. Mathematically, symmetry and curvature are two very different descriptors of shape. However, the symmetry-curvature duality theorem proved in [14] shows that there is an intimate relationship between these two descriptors. Indeed the theorem states that, to each curvature extremum, there is a unique symmetry axis terminating at that extremum. Furthermore the Interaction Principle [14] statesthat each of the axes is a direction along which a process has acted. The implication is that the boundary was deformed along the axes; e.g. each protrusion was the result of pushing out along its axis, and each indentation was the result of pushing in along its axis. There are four alternative labels (*M*+, *m*−, *m*+, and *M*−) that could be associated to each axes and these correspond to the four alternative types of curvature extrema. *M* and *m* denote, respectively, a curvature maximum and minimum; and **+** and **−**, respectively, denote positive and negative curvatures.

The introduction of the Leyton'sprocess-grammar within the proposed approach allows identifying the exact position of the curvature extrema. Indeed the next picture shows that the curvature graph, calculated in the previous section, can be enriched of the information obtained through the Leytonprocess-grammar; in particular the highest curvature's value is $k(u) = 1{,}52$for $u = 7{,}71$.

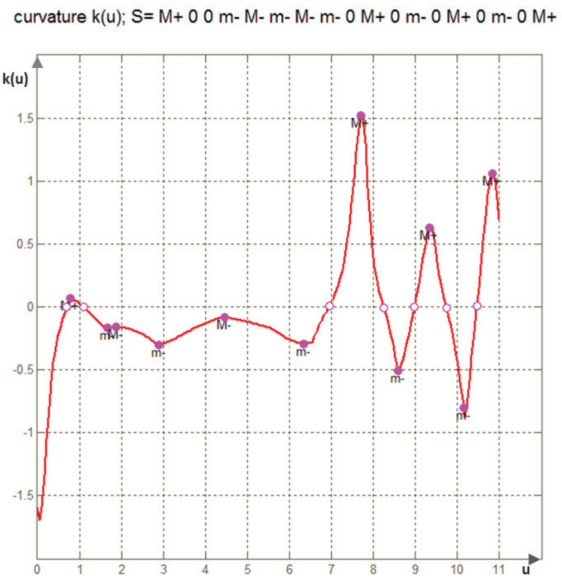



***Fig.7Curve's signature and parametric position of its extrema.***

As depicted in figure7, each curve's extremumhas been individuated with a label that specify the type of the extremum, and for each of themit is possible to detect: the parametric position on the curve, the value of the curvature and the type of the extremum.

## *3.3 Splitting Toolbox*

Before proceeding with the enforcement of shape constraints in the fitting problem it is necessary to increase the flexibility of the initial fitted B-Spline curve. This operation allows the engineers to restrict the influence of constraints in specific regions of the B-Splinecurve,from there it will be possible to impose various constraints in the fitting problem by specifying their type and range of influence.

For B-Spline editing the literature proposes different and various algorithms [27] that allow increasing or reducing the flexibility of a curve, these algorithms can be classified in: degree elevation/reduction; knot insertion; knot removal; reparameterization. The basic idea behind these techniques is to increase the flexibility of the curve without changing the shape of the curve. The success of the idea depends on the fact that there are an infinite number of control points with more than the minimum number of vertices that represent identical B-Spline curves [28].

In order to develop a toolbox that allows splittinglocally a B-Spline curve,it is necessary to enhance the flexibility of the B-Spline by increasing the number of the control points, which corresponds in inserting additional knot values into the knot vector$\boldsymbol{U}$ (*knot insertion* technique). The finalresult is a curve defined by a larger number of control points, but which defines exactly the same curve as before the knot insertion. This technique provides more local control by isolating a region to be modified from the rest of the curve, which thereby becomes immune from the local modification. In particular the development of the splitting toolbox is based on the *Boehm's knot insertion* algorithm [29, 30]. Consider the original curve ***B(u)*** defined by eq.1 with the knot vector$[U] = [u_1;\ u_2 \dots\ u_{n+p+1}]$. After knot insertion, the new curve is ***C(t)*** defined by:

$$\boldsymbol{C}(t) = \sum_{j=0}^{m-1} \overline{N}_{j,p}(t) \cdot P_j^* \qquad (6)$$

with the knot vector $[U^*] = [u_1^*;\ u_2^* \dots\ u_{m+p+1}^*]$ where *m* > *n*. The Boehm's algorithm allows to determine the new control polygon vertices $\boldsymbol{P}_j^*$ such that $\boldsymbol{B}(u) = \boldsymbol{C}(t)$.

In a nutshell, the implementation of the *splitting toolbox* allows inserting$\boldsymbol{G^0}$ continuities in the characteristic points detected during the *curvature analysis*; from a mathematical points of view, the outputs of the splitting toolbox are:

1. the modified knot vector $[U^*]$ with internal multiplicities;
2. the new set of $\bar{u}_k$ parameters (their number is equal to the number of control points $\boldsymbol{P}_j^*$) necessary to build the final bar network.

These outputs represent the data necessary in the last stage to perform the constrained force density fitting that allowsfinding the final shape of the B-Spline curve.

The next figure shows the result of the splitting toolbox applied to the B-Spline depicted in fig.8, the curve has been split in the point according the curvature analysis represented in fig.7. The $G^0$continuity has been obtained with a multiple internal knot that yields 3 new control points, one of which is exactly on the curve at the location $\boldsymbol{B}(u = 7{,}71)$:

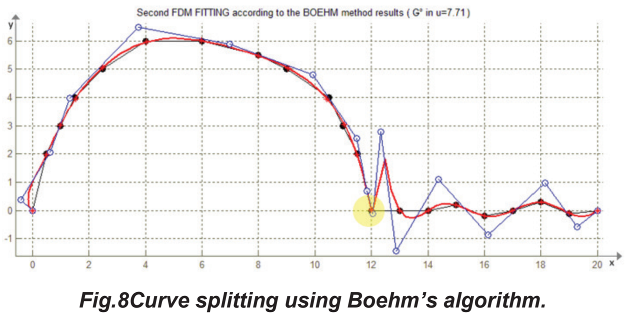



***Fig.8Curve splitting using Boehm's algorithm.***

## *3.4 Constrained Fitting*

As stated before, the method ends with the computationof the final constrained fitting curve taking into account: the set of points $\{Q_k\}, k = 1, \dots, m,$ to fit; the curvature information and the characteristic points of the initial fitted B-Spline curve inferred by the curvature evaluation toolbox; the modified knot vector $[U^*]$and the new set of $\bar{u}_k$ parameters output of the splitting toolbox.

While the first fitting toolbox yields a uniform B-Spline curve, in the last fitting procedure a non-uniform B-Spline curve has been adopted in order to utilize the output of the curvature analysis and splitting toolbox and then perform a constrained fitting curve. The non-uniform B-Spline model is more general than the uniformB-Splines curves, although it is not the most general type of this curve. It is obtained when the knot values are not equally spaced. The only requirement is that the knots have to be non-decreasing.

From a mathematical point of view,ourconstrained fitting problem is translated into the following optimization problem:

$$\begin{cases} min\,\Phi(F) = min \sum_k f_k^2 \\ G(F) = 0 \end{cases} \tag{7}$$

where$\Phi(F)$ is the objective function to minimize whose computation depends on the adopted mechanical model of bar network, and $G(F)$ is the set of shape constraints to enforce in the fitting problem.

The condition of the final fitting problem has been implemented using the Matlab function *"fmincon"* (find minimum of constrained nonlinear multivariable function) belonging to the *"Optimization ToolBox"* that allows to program linear/non-linear and equality/inequalityconstraints.As stated before, only some shape constraints have been programmed and tested, such as: position, distance, straight and curvature constraints.

***Position Constraint***. The position constraint is a linear equality constraint that can be imposed eitheron one or more control points $\boldsymbol{P}_i$ oron point/s of the B-SplinecurveB$\boldsymbol{B}(u)$. In the first case two equations are added to the system for each constrained control point. The equations of each position constraint added to the fitting problem are calculated automatically by the correspondent function implemented in MatLab, which accepts the $(x_i, y_i)$coordinates of the control point to constrain as input parameters.

Also the position constraint applied on a specific point of the B-Splinecurve yields two linear equality equations to add to the system. For example, the introduction of a position constrainton the specific point $\boldsymbol{B}(u = 0{,}5) = (x_s, y_s)$can be written as:

$$\begin{cases} B_x(u = 0{,}5) = \sum_{i=0}^{n-1} N_{i,p}(u = 0{,}5) \cdot x_i = x_s \\ B_y(u = 0{,}5) = \sum_{i=0}^{n-1} N_{i,p}(u = 0{,}5) \cdot y_i = y_s \end{cases} \tag{8}$$

***Distance Constraint.*** The distance constraint is an equality non-linear constraint that can be applied either betweentwo control points of the B-Spline curve, i.e. $\|\boldsymbol{P}_j - \boldsymbol{P}_i\| = d$, orbetween two points of the curve, i.e.$\|\boldsymbol{B}(u_i)\boldsymbol{B}(u_j)\| = d$.In both cases the imposition of a distance constraint implies the addition of one equation to the system:

- $$(P_{xi} - P_{xj})^2 + (P_{yi} - P_{yj})^2 - d^2 = 0 \tag{9}$$
- $$(P_{xi} - P_{xj})^2 + (P_{yi} - P_{yj})^2 - d^2 = \left(\sum_{i=0}^{n-1} N_{i,p}(u_i) \cdot x_i - \sum_{i=0}^{n-1} N_{i,p}(u_j) \cdot x_i\right)^2 + \left(\sum_{i=0}^{n-1} N_{i,p}(u_i) \cdot y_i - \sum_{i=0}^{n-1} N_{i,p}(u_j) \cdot y_i\right)^2 - d^2 = 0 \tag{10}$$

***Straight Constraint.*** In order to impose that a specific region of the B-Splinecurve has to be straight it is necessary to apply the straight constraint on the control points $\boldsymbol{P}_i$ of the curve.Such a constraint can be expressed with the followingvectorial equation:

$$(\boldsymbol{P}_i - \boldsymbol{P}_{i-1}) \wedge (\boldsymbol{P}_{i+1} - \boldsymbol{P}_i) = \boldsymbol{0} \tag{11}$$

that is an equality non-linear constraint. In order to introduce automatically this type of constraint,in the fitting problem, the function implemented in MatLabaccepts the coordinates of the control points as input parameters.

***Curvature Constraint.*** The curvature constraint is an equality non-linear constraint that can be formulated using the subsequent equation:

$$\left| \frac{\dot{B}_x(\bar{u}_k) \cdot \ddot{B}_y(\bar{u}_k) - \dot{B}_y(\bar{u}_k) \cdot \ddot{B}_x(\bar{u}_k)}{\left(\left(\dot{B}_x(\bar{u}_k)\right)^2 + \left(\dot{B}_y(\bar{u}_k)\right)^2\right)^{3/2}} \right| = \left(\frac{1}{r}\right) \tag{12}$$

where$\bar{u}_k$ is the set ofpoints of the B-Splinecurve on which the curvature constraint are imposed, and $r$ is the curvature's radius. The derivatives $\dot{\boldsymbol{B}}(u)$ and $\ddot{\boldsymbol{B}}(u)$ of a B-Spline curve at any point on the curve are obtained by formal differentiation. Specifically, the first and the second derivative are, respectively:

$$\dot{\boldsymbol{B}}(u) = \sum_{i=0}^{n-1} \dot{N}_{i,p}(u) \cdot \boldsymbol{P}_i \ ; \ \ddot{\boldsymbol{B}}(u) = \sum_{i=0}^{n-1} \ddot{N}_{i,p}(u) \cdot \boldsymbol{P}_i \tag{13}$$

The derivatives of the basis functions (eq.2 and eq.3) are also obtained by formal differentiation:

$$\dot{N}_{i,p}(u) = \frac{N_{i,p-1}(u) + (u - x_i)\dot{N}_{i,p-1}(u)}{x_{i+p-1} - x_i} + \frac{(x_{i+p} - u)\dot{N}_{i+1,p-1}(u) - N_{i+1,p-1}(u)}{x_{i+p} - x_{i+1}} \tag{14}$$

Differentiating the previous equation yields the second derivative of the basis function:

$$\ddot{N}_{i,p}(u) = \frac{2\dot{N}_{i,p-1}(u) + (u - x_i)\ddot{N}_{i,p-1}(u)}{x_{i+p-1} - x_i} + \frac{(x_{i+p} - u)\ddot{N}_{i+1,p-1}(u) - 2\dot{N}_{i+1,p-1}(u)}{x_{i+p} - x_{i+1}} \tag{15}$$

For example, let us suppose we are performing a constrained fit that starting from the initial fitting (fig.5) and the information inferred from the curvature analysis (fig.7) and the splitting toolbox (fig.8) consider also a curvature and a straight constraint, the following figure shows the resultant final constrained FD B-Spline fitted curve:

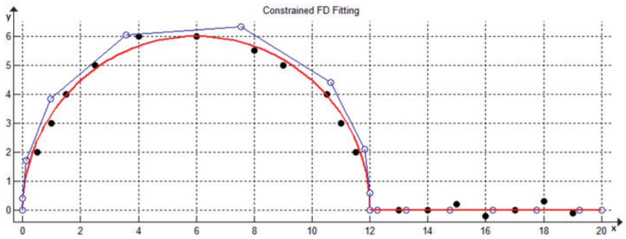


*Fig.9 Final constrained fitted curve.*

In this example we have imposed $r = 6$but, as general rule, the curvature's radius is a well-known parameter reliant on the product or the technological process however it may also be inferred from the curvature plot.

## 4 Conclusion

The paper has proposed a new approach for a constrained B-Spline fitting technique based on the Force Density Method. Its usefulness has been illustrated with some shape constraints where the enforcement, in the fitting problem, may be necessary to satisfy tolerances and reproduce regularities required by technological processes and engineering applications.

The method is composed by four stages: for each one, different functions have been implemented in MatLab in order to automate the whole process. The method has been validated with the implementation of simple shape constraints, but the functionalities present in the Optimization Toolbox, adopted as a solver, allow to develop any kind of constraint and then to create libraries of shape constraints depending on the engineering applications or technological processes to be adopted.

The results suggest that the adoption of a mechanical model of barnetworks (FDM) allows developing a more flexible tool than the traditional LSMs usually adopted for fitting problems. Indeed, as described in the paper, for specific values of the force densities the approach yields the same results of the LSM.

However there are some limitations due to the formulation of the B-Spline curve:B-Spline are still polynomial curves indeed, hence cannot represent many useful simple curves such as circles and ellipses. Nevertheless this approach could be extended and easily readapted for the NURBS curves,which allow to further enhance the flexibility of the proposed technique.

## References


[1] S. L. Arlinghaus. *Practical handbook of curve fitting*. CRC Press, 249pp, 1994.

[2] N.Chernov,C.Lesort, N.Simányi.*On the complexity of curve fitting algorithms.* Journal of Complexity, v.20 n.4, August, 2004,pp.484-492.

[3] N.Chernov,C.Lesort.*Statistical efficiency of curve fitting algorithms*. Computational Statistics & Data Analysis, 2004,pp. 713-728.

[4] J. Yang, H.Byun.*Curve fitting algorithm using iterative error minimization for sketch beautification*. 19th International Conference on Pattern Recognition, 2008. ICPR 2008.

[5] J. Hoschek.*Intrinsic parameterization for approximation.* Computer Aided Geometric Design 5, 27–31. 1988.

[6] A. Atieg, G.A. Watson. *A class of methods for fitting a curve or surface to data by minimizing the sum of squares of orthogonal distances.* Journal of Computational and Applied Mathematics 158, 227–296. 2003.

[7] W. Wang, H. Pottmann, Y. Liu. *Fitting B-Spline curves to point clouds by curvature-based squared distance minimization.* ACM Transactions on Graphics 25, 214–238. 2006.

[8] A. Blake, M. Isard, M. *Active Contours*. Springer, New York. 1998.

[9] H.J. Schek.*The force density method for form finding and computation of general networks*. Comp. Meth. in App. Mech. and Eng., vol. 3(2), 1974,pp. 115–134.

[10] L.Gründig,E.Moncrieff, P. Singer and D. Ströbel. *A history of the principal developments and applications of the force density method in Germany 1970-1999*. In Papadrakakis Ed., Proc. IASS-IACM 2000, June 2000, Chania-Crete, Greece.

[11] P.Véron, P.Trompette and J.C. Léon.*Integrated design and collaborative engineering of fabric structures*. Engineering with Computers, vol. 14, 1998,pp. 23–35.

[12] S. Guillet,J.C. Léon. *Deformation criteria for the direct manipulation of free form surfaces*. Curves & Surfaces, edited by L. Schumaker and A. Le Mhaut, Saint-Malo, 2000.

[13] J.P. Pernot,S. Guillet, J.C. Lèon, B. Falcidieno, F. Giannini.*Interactive operators for Free Form Features manipulation*. Book of the SIAM Conference on Geometric Design & Computing, 10-13 November 2003, Seattle, USA.

[14] M. Leyton.*Symmetry-curvature duality*. Computer Vision, Graphics, and Image Processing, 38, 1987, pp.327-341.

[15] URL: http://www.mathworks.com/

[16] E.Saux. B-Spline Curve Fitting: Application to Cartographic Generalization of Maritime Lines. International Conference Graphicon 1998, Moscow, Russia.

[17] A.Ruas, J.P. Lagrange.*Etat de l'art en généralisation& survey on generalization.* Technical Report DT-93-0538, IGN-France, 1993.

[18] J.C. Léon, P. Trompette. *A new approach towards free-form sufaces control*. Computer Aided Geometric Design, vol. 12, 1995,pp. 395–416.

[19] J. Sanchez, M.A. Serna,P. Morer. *Using NURBS Fitting Techniques To Represent Tensile Structures*. Adaptables2006, TU/e, International Conference On Adaptable Building Structures. Eindhoven The Netherlands 03-05 July 2006.

[20] D.F. Rogers, N.G. Fog. *Constrained B-Spline Curve and Surface Fitting*. Computer Aided Design, 21[10], 1989, pp.641-648.

[21] L.Piegl, W.Tiller. *The NURBS Book*. Springer, New-York, 2nd edition, 1997.

[22] MatLab documentation, URL: http://www.mathworks.com/help/toolbox/optim/

[23] C.De Boor. *A Practical Guide to Splines*. New York: Springer-Verlag,1978.

[24] D.Salomon. *Computer Graphics and Geometric Modeling.* New York, Springer-Verlag,1999.

[25] D.Salomon. *Curves and surfaces for computer graphics*.New York, Springer-Verlag,2006.

[26] M. Leyton.*Process Grammar.* DIMACS Technical Report 2007-09, Rutgers University, New Brunswick, NJ 08854, September, 2007.

[27] L.Piegl, W. Tiller. *Algorithm for degree reduction of B-Spline curves*. CAD, Vol. 27, No. 2, 1995,pp. 101-110.

[28] D.F. Rogers.*An Introduction to NURBS: With Historical Perspective*.Morgan Kaufmann publisher, San Diego, 2001.

[29] W. Boehm.*Inserting new knots into B-Spline curves*. CAD, Vol. 12, 1980,pp. 199-201.

[30] W. Boehm,H.Prautzsch. *The insertion algorithm*. CAD, Vol. 17, 1985,pp. 58-59.