ELSEVIER

Computer-Aided Design 38 (2006) 584–594

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

# Automatic surface modelling of a ship hull

F. Pérez-Arribas *, J.A. Suárez-Suárez, L. Fernández-Jambrina

*Ensenanzas Basica Ingenieria Naval, Naval Architecture School of Madrid, Universidad Politécnica de Madrid,*
*Avenida Arco de la Victoria SN, 28040 Madrid, Spain*

Received 22 July 2005; accepted 28 January 2006

**Abstract**

When defining a ship hull surface, the main objective is to obtain a faired surface or surfaces that contain some specific points of the hull, that have been selected in the design process and give the ship its hydrodynamic, stability and other properties. So, the hull surface should be a compromise between fairness and precision, and this is not and easy task. In this paper, authors present a thorough procedure for automatic modelling with a fair NURBS surface, having lists of points on the stations of the vessel as initial data. The construction of spline curves, and their application in the definition of ship lines is reviewed. Approximation of spline curves fitting the data on the stations is made, with special emphasis on the choice of parametrization, which is relevant to increase the accuracy of the splines. NURBS surfaces modelling the hull of the vessel are constructed and the fairing process adapted to maintain certain ship characteristics is described.

*Keywords:* Shape generation; Ship surface design; Design validation; Surface fairing

## 1. Introduction

Fitting and fairing the lines drawing from the data of ship offsets is often the beginning of a complete ship hull design procedure. This means to define the surface of the ship hull, which contains or better approximates these offsets or points belonging to the hull. These tasks may involve a considerable amount of time and are usually a bottleneck in the process of design, whereas their result is essential information for subsequent processes. Any surface modelling system for naval architecture should fulfil a number of requirements:

- Easy and quick fitting of data within a predetermined tolerance level.
- Quick and intuitive fairing methods which, although grounded on complex algorithms, may prevent the user from involving in complicated mathematical calculations.

One of the most important characteristics of a CAD system is its capability of exchanging information with other systems in a large range of formats. The input information in a fitting process is usually given through ship offsets and therefore it is necessary to read files with numerical information (specified points of the ship hull) in text format. Concerning the output of the modelling system, this is usually in DXF format if we are interested in just fitting stations into spline curves (wire models, Fig. 1, up), or in IGES format if we want one or several NURBS surfaces constructed with these data (surface model, Fig. 1, down). The latter one is the most common format for exchanging information with computer fluid dynamics (CFD) and finite element (FEM) programs.

The first step towards construction of the ship hull is plotting curves that comprise the geometrical information of the surfaces that are aimed to obtain. That is, we want to convert in a quick and efficient way a set of points in a number of splines, which contain all relevant information for constructing the surface. Concerning the fitting process, the method should provide a measure of the error that has been made in order to comply with predetermined tolerances (in naval architecture they are usually of the order of a few millimeters, even centimeters; this is a function of the construction process, of the material and of the ship size).

Usual ship files for an ordinary vessel contain between 11 and 25 stations (hull sections as in Fig. 1, up) and 8–10 waterlines (hull sections parallel to the water surface). This means at least 100 intersection points. When using NURBS, an exact interpolating surface would require a control net with such large number of control points, which would make it difficult to handle in case of requiring a fairing process, since each control point acts very locally on the curve or surface.

* Corresponding author. Tel.: +34 913 367204; fax: +34 915 442149.
*E-mail address:* fperez@etsin.upm.es (F. Pérez-Arribas).

0010-4485//$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2006.01.013

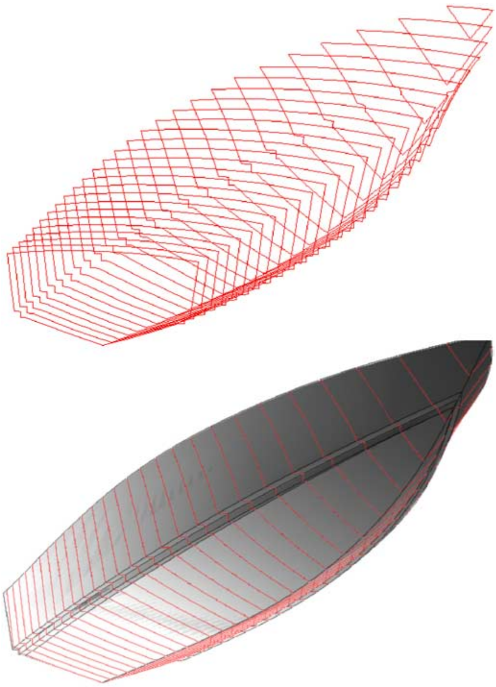


Fig. 1. Wire and surface models.

Therefore, and besides for simplicity, it is convenient to use an approximation method with convenient precision in order to fit the forms of a vessel.

The hull of a ship can be a complex surface, specially at the bow and stern parts. There are nearly no two identical ships and therefore generation and definition of forms is almost a labour of an experienced naval architect, though the use of computers makes the task easier. In spite of this, design with plastic or metal splines, or loftsman's splines, is still quite extended when working with a printed design.

A physical spline is a long, narrow strip of wood, plastic or metal used to fit curves through specified data points [1]. The splines are shaped by lead weights called 'ducks' (Fig. 2) and by varying the number and position of the weights the spline is made to pass through the specified data, as smooth or fair as possible.

The use of surfaces for modelling the hull of a ship is not just aesthetical. The forms of a ship must be fair or smooth, that is, they should not show 'bumps' which would affect the hydrodynamic behaviour of the ship [2], increasing ship resistance. In a wire model, one should represent one by one the curvature of the stations or of the waterlines and in case they show fairness problems, more sections of either kind should be inserted in the problematic zone and modified accordingly in order to smooth the curvature. From the construction point of view, the use of surfaces enables the interpolation of lines on ship hull easy and accurately, which are necessary to build the internal ship structure.

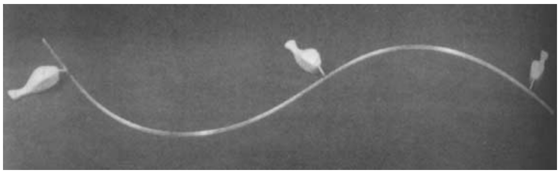
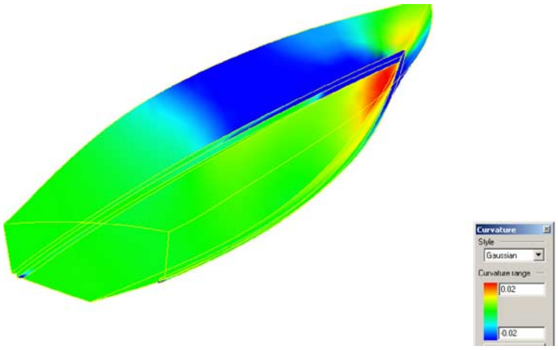


Fig. 2. The loftsman's spline.



Curvature
Style
Gaussian
Curvature range
0.02
-0.02

Fig. 3. Curvature representation.

If the hull surface is defined, total curvature may be obtained mathematically (Fig. 3) in order to detect critical zones and act on them, as we shall see later on. This is specially critical in regions of high curvature, such as the bow. Furthermore, in case of having a wire model, if we wish a new station, interpolation for such curve is not easy, specially if the stations show points called knuckles, where the continuity of the curvature is lost, as it happens in Fig. 1, and the union of the different surfaces in these knuckle lines is $C^1$.

The main disadvantage of the use of surfaces for modelling the hull of a ship is that they are difficult to control. With an interpolating wire model one is sure that the hull will pass through the points on the stations. On drawing approximating surfaces with such stations, the section along a plane containing the station will not provide exactly the same curve (tolerances are always allowed), due to the properties of surfaces that we will explain later on.

Therefore, the aim of this paper is to present a thorough procedure for automatic modelling with a fair NURBS surface, having lists of points on the stations of the vessel as initial data. The paper is organized as follows. In this section, we review the construction of spline curves, and their application in the definition of ship lines.

Section 2 is devoted to approximation of spline curves fitting the data on the stations, with special emphasis on the choice of parametrization, which is relevant to increase the accuracy of the splines. In Section 3, NURBS surfaces modelling the hull of the vessel are constructed and the fairing process is depicted in Section 4. The results are discussed at the end of the paper.

### 1.1. What is a NURBS?

The surfaces that best model the hull of a ship are non-uniform rational B-spline (NURBS) surfaces. In order to introduce the notation for the paper, we review briefly how they are produced (cf. for instance [3–6]).

A B-spline curve is formed by several pieces of polynomial or rational curves and the whole curve is $C^2$ at the junctions, in the case of cubic B-splines. It is defined by a polygon called control polygon, and an interpolation algorithm that allows the construction of the curve relating the curve to the control polygon. The interpolation steps are encoded in a family of piecewise polynomial functions $N_j^n(u)$ called B-spline functions of degree $n$. The order of the functions is $n+1$ and stands for the number of non-null pieces that the B-spline functions may have. In Fig. 5, a cubic B-spline is shown (degree 3, order 4). Three is the most usual degree in ship design and the one that fits better the traditional loftsman's splines.

A spline curve is a linear combination of B-spline functions with $m+1$ control points as coefficients. So, spline curves are parametric, $x=g(u)$, $y=h(u)\ldots$ In the plane, $V_i=(X_i, Y_i)$, $i=0,\ldots,m$, generate a spline $s(u)$ of degree $n$,

$$s(u) = \sum_{j=0}^{m} V_j \cdot N_j^n(u) = (X(u), Y(u))$$

$$= \sum_{j=0}^{m} (X_j \cdot N_j^n(u), Y_j \cdot N_j^n(u)) \tag{1}$$

Rational curves may also be defined,

$$s(u) = \frac{\sum_{j=0}^{m} w_j \cdot V_j \cdot N_j^n(u)}{\sum_{j=0}^{m} w_j \cdot N_j^n(u)} \tag{2}$$

by introducing a set of numbers, $w_j$, $j=0,\ldots,m$, called weights. If all of them are one, the polynomial B-spline is recovered, since a nice property of B-spline functions is that their sum is always unity,

$$\sum_{j=0}^{m} N_j^n(u) = 1$$

for all values of the parameter $u$.

In the case of a ship, the use of parametric coordinates is essential, since there are stations on the vessel that have more than one ordinate for the same abscissa, as is happens in regions close to the bow, Fig. 4, and therefore these stations cannot be drawn as functions of single-valued functions of type $y=f(x)$.

The parameter $u$ grows monotonically from one endpoint of the curve to the other. It is usually normalized to take values between zero and one, but this is not necessary. It bears no simple relation to the length of the curve. The price that has to be paid for using parametric coordinates is that the inverse relation, that is the relation which provides $u$ in terms of $x$ or $y$ is not simple and therefore it is difficult to know the value of $u$ for a given point of the ship offsets.

In Fig. 5 left, the vertexes of the control polygon of a spline that models a station of the bow of a vessel are numbered from 0 to 5. The straight lines that link them are the sides of the control polygon. It may be seen that the shape of the polygon mimics the shape of the curve and therefore provides a rough estimation of it. As it is well known, by moving the control points the shape of the curve is altered.

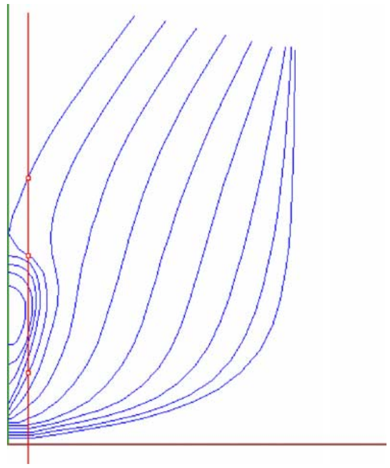


Fig. 4. Parametric coordinates in ship stations.

In addition to the control polygon, a spline curve has also a list of knots, which are the values of the parameter $u$ at the junctions between pieces. We shall refer with the word knot either to the junction point or the value of the parameter at the junction. In Fig. 5 right, the knots are marked with dots. They are usually taken equally spaced in $u$ along the curve, although this is not strictly necessary. The values of the knots in Fig. 5 are $u=0$, 1/3, 2/3 and 1.

Mathematically, the knots just mark the beginning and the end of the pieces of the spline curve, three in Fig. 5. Physically, these are the points of the loftsman's spline where the weights were set to modify their curvature when drawing curves (Fig. 2). Changing the position of these weights, causes the shape of the curve to be changed in order to fit the data points.

The B-spline curve is forced to pass through the first and last vertex of the control polygon, corresponding to knots $u=0$ and 1. The first and last sides of the control polygon provide the direction of the tangents to the curve at the endpoints. This is achieved by repeating the knots at $u=0$ and 1 on the spline.

The number of knots equals the number of control points plus the order of the curve. In Fig. 5, there should be 10 knots and in Fig. 6, eight knots. In Fig. 6, for a cubic spline the list of knots could be [0, 0, 0, 0, 1, 1, 1, 1], that is, the first knot is repeated four times, as much as the order of the spline, and so is the last knot. In principle, three times would be enough, since the first and last knots are not necessary for calculation of the spline, although they appear in IGES formulation. As it has already been said, the knots are usually taken equally spaced. In Fig. 5, the list of knots could be [0, 0, 0, 0, 1/3, 2/3, 1, 1, 1, 1]. These lists of knots with non-trivial multiplicity are called open sequences of knots.

### 1.1.1. Calculation of the B-spline functions

A spline curve of degree $n$ is a linear combination of B-spline functions of the same degree. These functions may be constructed recursively from lower to higher degree in terms of

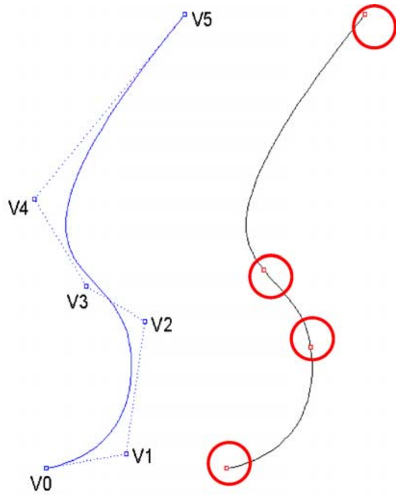


Fig. 5. Control points and knots for a B-spline.

the list of knots, starting at $u_{-1}$. These basis functions can be calculated with the De Boor algorithm of Eq. (3)

$$N_j^0(u) = \begin{cases} 1 & u \in [u_{j-1}, u_j) \\ 0 & u \notin [u_{j-1}, u_j) \end{cases}$$

$$N_j^n(u) = \frac{u - u_{j-1}}{u_{j+n-1} - u_{j-1}} \cdot N_j^{n-1}(u) + \frac{u_{j+n} - u}{u_{j+n} - u_j} \cdot N_{j+1}^{n-1}(u) \qquad (3)$$

## 2. Mean square approximation of stations with a cubic spline

As previously mentioned, each station will be given by a list of data points of the ship hull or offsets, formed by $p+1$ points $P_i=(x_i, y_i)$, $i=0,\ldots,p$ $(3<m<p)$, necessarily enclosing the first and last points of the curve, through which the spline will pass. We wish to find a list of $m+1$ control points $V_i=(X_i, Y_i)$, $i=0,\ldots,m$ $(3<m<p)$, which defines the $n$-degree spline of parametric equation $s(u)$ which is closer to the data points, according to Eq. (1), considering the least square fitting criteria.

We choose cubic splines ($n=3$) and not another degree for simplicity of their formulation and for their nice properties (i.e. possibility of an inflection point in each piece, class $C^2\ldots$). We have also mentioned their similarity with the curves drawn with the traditional design techniques.

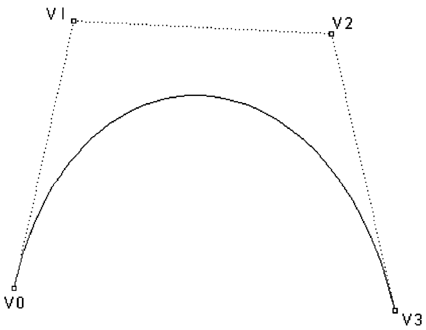


Fig. 6. Construction of a spline curve.

Besides, cubic splines mathematically approximate the previously mentioned physical spline of Fig. 2, a flexible strip of wood with a number of adjustable weights that was formerly used for drawing hull forms. In Fig. 7, there appear some stations, fitting with splines the actual points, shown as circles.

Spline curves also fulfil that $n+1$ aligned control points, where $n$ is the degree of the spline, produce a straight piece of line. This feature is used to define straight regions of certain stations with a single spline, which are normally used in ship hull design because these parts are easy to construct with steel plates.

### 2.1. Choosing the list of knots

The spline must pass through the first and last point of the data set. Therefore, the first and last control points must be the endpoints of the original curve and multiplicity three (four, counting the additional knots) will be assigned to the parameter values corresponding to these points. We may choose the remaining knots arbitrarily. They are usually taken equally spaced between 0 and 1 or spaced with constant difference equal to unity,

$$u-1 = u_0 = u_1 = u_2 = 3,\ u_3 = 4, \ldots, u_m = u_m + 1$$
$$= u_m + 2 = u_m + 3 = m + 1$$

Once the approximating spline has been defined, we have to choose a function that measures the distance between the spline and the original data. The euclidean distance between actual and approximating points may be used to define

$$R = \sum_{i=0}^{p} |s(U_i) - \mathbf{P_i}|$$
$$= \sum_{i=0}^{p} [(X(U_i) - x_i)^2 + (Y(U_i) - y_i)^2] \qquad (4)$$

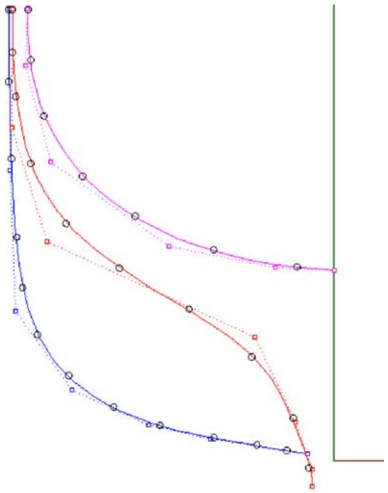


Fig. 7. Stations modelled as cubic B-splines.

where $U_i$, $i=0,\dots,p$, is the value of the parameter $u$ associated with the point $P_i$ of the data set. This point is related to a point in the spline, $s(U_i)$, and therefore $R$ measures the distance between the spline and the original station.

### 2.2. Choosing a parametrization

The value of function $R$ will decrease as the spline points approximates to the original data set. For this premise, the choice of parameters $U_i$, $i=0,\dots,p$, is determinant. This is called the choice of parametrization. There are many ways of choosing the parametrization. The most usual methods are the uniform parametrization, the parametrization by chord length and the centripetal parametrization [4,7].

In a first approximation we will use the centripetal parametrization to obtain the values $U_i$, $i=0,\dots,p$, since it usually provides a nicer fitting for a list of points,

$$U_i - U_{i-1} = k\cdot\sqrt{|\mathbf{P_i}-\mathbf{P_{i-1}}|},\quad i=1,\dots,p \tag{5}$$

where the initial value $U_0$ and $k$ depend on the origin and width of the interval of definition of the parameter $u$, in this case, respectively, 3 and $m+1$,

$$U_0 = 3,\quad k = \frac{m-2}{\sqrt{|\mathbf{P}_1-\mathbf{P}_0|}+\cdots+\sqrt{|\mathbf{P}_p-\mathbf{P}_{p-1}|}}$$

### 2.3. Solving the approximation problem

In order to obtain the cubic spline with $m+1$ control points, which is closer to the data, we minimize the distance function $R$, using as free parameters the coordinates of the control points, except for the first and last points, which are already determined. We are left with $m-1$ equations for each variable, $X$, $Y$,

$$\sum_{j=0}^{m}\left(\sum_{i=0}^{p} N_k^3(U_i)\cdot N_j^3(U_i)\right)\cdot X_j = \sum_{i=0}^{p} x_i\cdot N_k^3(U_i) \tag{6}$$

$$\sum_{j=0}^{m}\left(\sum_{i=0}^{p} N_k^3(U_i)\cdot N_j^3(U_i)\right)\cdot Y_j = \sum_{i=0}^{p} y_i\cdot N_k^3(U_i)$$

We shall call $B$ the matrix formed by $b_{ij}=N_j^3(U_i)$, $i=0,\dots,p$, $j=0,\dots,m$

$$B = \begin{pmatrix} N_0^3(U_0) & \cdots & N_m^3(U_0) \\ \vdots & \ddots & \vdots \\ N_0^3(U_p) & \cdots & N_m^3(U_p) \end{pmatrix}$$

The system of Eq. (6) may be written in matrix form as

$$B^tBX = B^tx \tag{7}$$

where $B^t$ denotes the transposed matrix of $B$, $b^t_{ij}=b_{ij}$, and $X$ and $x$ are, respectively, the matrices of coordinates of both the control points and the data points,

$$X = \begin{pmatrix} X_0 & Y_0 \\ \vdots & \vdots \\ X_m & Y_m \end{pmatrix},\quad x = \begin{pmatrix} x_0 & y_0 \\ \vdots & \vdots \\ x_p & y_p \end{pmatrix}$$

Since the first and last control points are known, because they are, respectively, equal to the first and last data points, the first and last columns of this system of equations are moved to the right-hand side. If we write the columns of the matrix $B$ as

$$B_i = N_i^3(U_0),\dots,N_i^3(U_p)$$

the matrix of the system of equations is provided by their scalar product, $C=(c_{ij})$, $c_{ij}=B_i\cdot B_j$, $i,j=1,\dots,m-1$,

$$C = \begin{pmatrix} \mathbf{B}_1\cdot\mathbf{B}_1 & \cdots & \mathbf{B}_1\cdot\mathbf{B_{m-1}} \\ \vdots & \ddots & \vdots \\ \mathbf{B_{m-1}}\cdot\mathbf{B}_1 & \cdots & \mathbf{B_{m-1}}\cdot\mathbf{B_{m-1}} \end{pmatrix}$$

This matrix is obviously squared and symmetric and therefore the system

$$C\cdot\begin{pmatrix} X_1 \\ \vdots \\ X_{m-1} \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{p} x_i\cdot N_1^3(U_i) - \mathbf{B}_1\cdot\mathbf{B}_0\cdot X_0 - \mathbf{B}_1\cdot\mathbf{B}_m\cdot X_m \\ \vdots \\ \sum_{i=0}^{p} x_i\cdot N_{m-1}^3(U_i) - \mathbf{B}_{m-1}\cdot\mathbf{B}_0\cdot X_0 - \mathbf{B}_{m-1}\cdot\mathbf{B}_m\cdot X_m \end{pmatrix}$$

$$C\cdot\begin{pmatrix} Y_1 \\ \vdots \\ Y_{m-1} \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{p} y_i\cdot N_1^3(U_i) - \mathbf{B}_1\cdot\mathbf{B}_0\cdot Y_0 - \mathbf{B}_1\cdot\mathbf{B}_m\cdot Y_m \\ \vdots \\ \sum_{i=0}^{p} y_i\cdot N_{m-1}^3(U_i) - \mathbf{B}_{m-1}\cdot\mathbf{B}_0\cdot Y_0 - \mathbf{B}_{m-1}\cdot\mathbf{B}_m\cdot Y_m \end{pmatrix}$$

has a unique solution. In this case, Gauss method is used to solve the system of equations of Eq. (7). For most ship hull forms, stations are modelled with no more than 10 control points, which can be managed pretty good with Gauss method.

### 2.4. Searching for the optimal parametrization

Although the centripetal parametrization usually provides nice results, it is desirable to obtain the best spline fitting for the data points of the ship hull with a minimum number of control points. We shall try to optimize the parametrization by an iterative scheme, using as initial seeds the values $U_i$, $i=0,\dots,p$, obtained with the centripetal parametrization.

This is one of the key points of the process, since the quality of the approximation of the station points obviously determines the quality of the hull approximating surface.

Once the approximating spline $s(u)$ is obtained with the values of $U_i$ provided by Eq. (5), we calculate the line perpendicular to the spline curve from each data point $P_i$. Such line cuts the spline at a point $I_i$, so that distance $P_iI_i$ is the minimum distance between the data point and the spline. Then, we take the value $U_i^1$ of the parameter $u$ corresponding to $I_i$, $I_i = s(U_i^1)$, obtaining a new family of parameters, $U_0^1, \ldots, U_p^1$.

We calculate now the parametrization for the approximating spline $s^1(u)$ with the new parameters, which provides a better precision fitting for the list of points $P_i$. Iterating this process $j$ times, we get a family $U_i^j$ of parameters for which the precision of the associated approximating spline $s^j(u)$ is increased several times compared to the original centripetal parametrization.

In order to calculate the error made in the approximation, we divide the spline in a large number of points (i.e. 5000) and the distance $P_iI_i$ is calculated numerically. In Fig. 8, it is shown the effect of the choice of parametrization in the fitting of the points of a station in the bow region, 6.35 m draught (maximum height of the station) and 1.7 m half breadth (maximum width of the station), with a spline of six control points. In Fig. 8, the parametrization is centripetal, the first step of approximation. With just one iteration (Fig. 8), the improvement is hardly perceptible, but in Fig. 8, 20 iterations have been carried out, 50 in Fig. 8, and it is shown that the error diminishes on increasing the number of iterations.

Fig. 8 shows the case of a bulbous bow. In the case of the examples of Section 4 that correspond to sailing ships, the error is reached with just two or three iterations and the presented example of Fig. 8 is more interesting to see the effect of the parametrization.

If a certain tolerance is imposed on the approximation, iterations are carried out until the maximal distance $P_iI_i$ becomes less than the tolerance value. If the tolerance cannot be attained no matter the number of iterations, the number of control points is to be increased.

Once this process is carried out on the list of points for every station (among them it is also possible to include the bow form profile), we have every station approximated with a cubic spline and also a measure of the approximation error made in each of them. Increasing the number of control points may diminish this error. The tolerance can be taken to be zero by prescribing a number of control points equal to the number of data points, though in this case we are not approximating but interpolating a spline through the data.

Interpolation is just a special case of approximation, but the presented method and the examples of Section 4 uses approximation instead interpolation, because this way more simple curves and surfaces can be obtained, which will facilitate the fairing process. This approximation gives good enough results for the tolerances normally used in ship design.

The iterative process used in the construction of the optimal B-spline curve for each station is a technique introduced by Hoschek [19]. Using direct least-square minimization over control points for each iteration, Hoschek's method remains computationally intensive and globally weak because problems of convergence due to rough parameter corrections that can be encountered. An improvement of Hoschek's method was proposed by Saux and Wallner [20] in order to get a more robust parametrization optimization step. For ship hull forms, that authors have tested, the parametrization method used in this work does not show convergence problems. Most complex parts of ship can be bulbous bow forms as the one of Fig. 8 and this forms are correctly solved with the parametrization process described in this work.

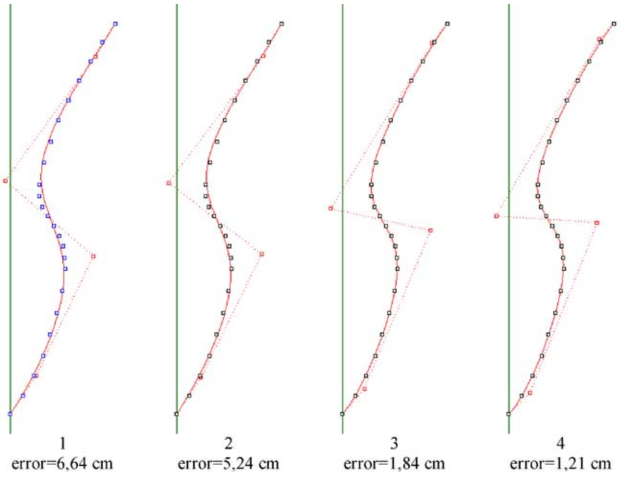


Fig. 8. Effect of the parametrization.

Now we have all the information needed to write a DXF file with the whole wire model of the ship.

### 2.5. Stations with straight pieces

The use of straight pieces in stations to form the flat bottom and the flat sides is common in high displacement ships. Straight pieces are always a problem for modelling with NURBS surfaces or splines, since one has to be careful to attain that they are actually straight. If these straight pieces are not controlled, the ship may have points under the base plane or the maximum breadth may be surpassed.

The method that has been described for fitting the optimal parametrization may mimic nicely straight pieces on stations (Fig. 9 left). The approximation is better if more points are introduced on the straight pieces of the station (Fig. 9 right), so these points will have more influence in the numerical fitting.

If one wishes to ensure that straight pieces are actually straight by drawing a station with just one cubic spline, it is necessary to modify the described method by placing four aligned control points on the chosen straight piece (bottom or side) and if the tangency points with the bottom and the side are known, they must be used as knots for the spline.

In Fig. 10, it can be seen the four points aligned with the directions of the bottom and the side of a station. The points $T$ are the tangency points of the bilge with the mentioned straight pieces.

In this way, just one spline is needed to model stations with straight pieces avoiding the modelling of the bottom and the side with straight lines and defining separately the bilge that is to be joined to these lines preserving the tangency. This simplifies the modelization of the ship with surfaces that support on the splines, which fit the points of the ship offsets. This is shown in Section 3.

## 3. Generation of a spline surface through the stations

The generalization of cubic spline curves to bicubic spline surfaces is almost straightforward. The control polygon is substituted for a control net depending on two indices, $V_{ij}$ ($X_{ij}$, $Y_{ij}$, $Z_{ij}$), as in Fig. 11. Products of B-spline functions in two variables $u$ and $v$ and two lists of knots $\{u_{-1}, \ldots, u_{m+3}\}$, $\{v_{-1}, \ldots, v_{n+3}\}$ are used,

$$\mathbf{s}(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} \mathbf{V}_{ij} \cdot N_i^3(u) \cdot N_j^3(v) \tag{8}$$

$u \in [u_2, u_m], \ v \in [v_2, v_n]$

For constant $u$ we obtain cubic spline curves in $v$ with $n$ control points. For constant $v$ we obtain cubic spline curves in $u$ with $m$ control points. Therefore, we may construct the surface from a 2D net of spline curves. This is specially useful for our purposes, since it allows us to draw a surface from previously designed stations of the vessel.

In practice this is easy, since it just requires another minimum squares fitting, this time with $n+1$ control points for each of the $m+1$ longitudinal rows of control points corresponding to the approximating splines for the stations.

These new control polygons constitute the control net of the approximating surface for the initial list of data points.

In this case the $m+1$ lists of points will not be coplanar as before, since the bow form is included. Therefore the fitting must be 3D, which does not mean any additional complication, since one just needs to add a third coordinate $z$ in space.

After finishing this calculation process, a text file in format IGES 5.0 is created where the relevant information for the construction of the bicubic spline surface through the stations is stored.

## 4. Fairing process

Once the surface is fitted with an adequate number of control points, it is usually faired interactively by direct manipulation of the calculated control net, taking into account curvature diagrams of the surface.

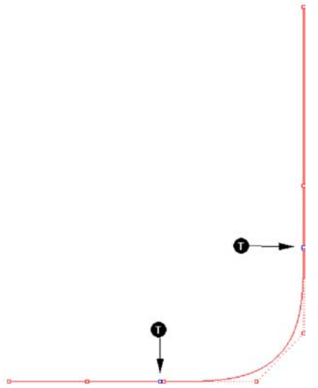


Fig. 10. Straight parts.

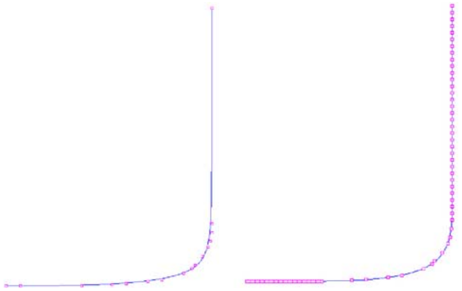


Fig. 9. Fitting stations with straight parts.

The main problem is that generally this fairing process may damage the level of precision acquired during the construction of the approximating surface, since the fairing and fitting process may interfere with each other. To avoid this we use an automatic fairing method that requires no direct participation of the user.

It is desirable that the fairing process should be local. If the surface needs to be faired only at a local spot due to the presence of an isolated bump, this should not mean that the whole surface has to be modified, and the hydrodynamic characteristics of the vessel will be maintained.

### 4.1. Fairing criterion

Among both lists of knots necessary to define the surface, we shall refer the inner knots as $u_3,\ldots,u_{m-1}$ and $v_3,\ldots,v_{n-1}$ and we assume that they all have multiplicity one. The whole set of indices for inner knots ($u_k$, $v_l$) is then

$I = \{(3,3); (3,4);\ldots;(m-1,n-1)\}$

If this criterion had to be fulfilled at every knot, it would mean that the spline surface would not be spline but polynomial, that is it would be a spline of just one piece. Since bicubic spline surfaces are generically $C^2$, their third-order derivatives are discontinuous at the knots. That is, we have just excluded the knots corresponding to the edges of the surface.

There are a certain number of fairing algorithms in the literature, but mostly for spline curves and not for surfaces.

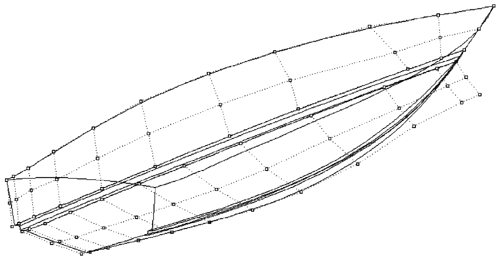


Fig. 11. Control point net.

Methods for splines curves are usually grounded on knot removal procedures [8–11]. However, the translations from curves to surfaces is far from straight forward.

We have chosen [12] for its simplicity and its local character, since it involves just nine control points in each iteration, compared with other alternative fairing algorithms [13,14,11]. We have modified [12] to maintain certain ship characteristics, as we will explain in the final comments of the fairing process.

Faired surfaces can be generated by means of minimization of certain fairness measures, which can be regarded as approximate strain energy [15]. A discussion about the selection of proper fairness measures can be found in [16] and [17]. In order to develop a fairing method we need a sort of quantitative measure of the fairness of the surface.

#### 4.1.1. Local fairness criterion

A $C^2$ spline surface $s(u, v)$ is fairer in a neighbourhood of the inner knot $(u_k, v_l)$ if $s(u, v)$ is locally $C^3$ at $(u_k, v_l)$.

Following this criterion, each local fairing iteration consists on reducing the differences between third-order partial derivatives at $(u_k, v_l)$. This means that fairing the whole surface amounts to reducing the value of the sum of third-order discontinuities at all inner knots on the surface.

A spline surface is $C^3$ at $(u, v)$ if and only if every third-order partial derivative of $s(u, v)$ is continuous at $(u, v)$. Since spline surfaces already have the nice property of having continuous third order cross partial derivatives, the sum of the differences along both $u$ and $v$ directions provides a reasonable measure of local fairness, according to the fairness criterion.

We may define discontinuity vectors at each knot,

$$\Delta_{uuu}(u_k, v_l) = \frac{\partial^3 \mathbf{s}}{\partial u^3}(u_k^-, v_l) - \frac{\partial^3 \mathbf{s}}{\partial u^3}(u_k^+, v_l) = \sum_{i=k-4}^{k} \sum_{j=l-3}^{l-1} \alpha_{ij} \cdot \mathbf{V}_{ij} \tag{9}$$

$$\Delta_{vvv}(u_k, v_l) = \frac{\partial^3 \mathbf{s}}{\partial v^3}(u_k, v_l^-) - \frac{\partial^3 \mathbf{s}}{\partial v^3}(u_k, v_l^+) = \sum_{i=k-3}^{k-1} \sum_{j=l-4}^{l} \beta_{ij} \cdot \mathbf{V}_{ij}$$

where generically the coefficients $\alpha_{ij}$, $\beta_{ij}$ (Table 1 and Table 2) depend on the knot $(u_k, v_l)$. But in the case of equally spaced inner knots, they take the same value for every knot,

Table 1
$\alpha_{ij}$ coefficients, $i = k-4,\ldots,k$, $j = l-3,\ldots,l-1$

| $\alpha_{ij}$ | $k-4$ | $k-3$ | $k-2$ | $k-1$ | $k$ |
|---|---|---|---|---|---|
| $l-3$ | $-1/6$ | $2/3$ | $-1$ | $2/3$ | $-1/6$ |
| $l-2$ | $-2/3$ | $8/3$ | $-4$ | $8/3$ | $-2/3$ |
| $l-1$ | $-1/6$ | $2/3$ | $-1$ | $2/3$ | $-1/6$ |

Table 2
$\beta_{ij}$ coefficients, $i = k-3,\ldots,k-1$, $j = l-4,\ldots,l$

| $\beta_{ij}$ | $l-4$ | $l-3$ | $l-2$ | $l-1$ | $l$ |
|---|---|---|---|---|---|
| $k-3$ | $-1/6$ | $2/3$ | $-1$ | $2/3$ | $-1/6$ |
| $k-2$ | $-2/3$ | $8/3$ | $-4$ | $8/3$ | $-2/3$ |
| $k-1$ | $-1/6$ | $2/3$ | $-1$ | $2/3$ | $-1/6$ |

Therefore, a local fairness measure $L_{kl}$ at the knot $(u_k, u_l)$ could be defined as

$$\mathrm{L_{kl}} = |\Delta_{uuu}(\mathrm{u_k}, \mathrm{v_l})|^2 + |\Delta_{vvv}(\mathrm{u_k}, \mathrm{v_l})|^2 \tag{10}$$

In the calculation of $L_{kl}$, 21 different control points take part.

As a whole, we may assign the whole surface $s(u, v)$ a quantity $G_s$ that may be called global fairness measure,

$$G_s = \sum_{(k,l) \in I} L_{kl} \tag{11}$$

Global fairness criterion: A bicubic spline surface $s(u, v)$ is fairer than another $s_0(u, v)$ if $G_s < G'_s$.

The strategy that we follow in order to improve the fairness of the surface involves two steps:

- A local fairing iteration at the knot $(u_k, v_l)$, $(k, l) \in I$, where $L_{kl} = \max(L_{ij})$, that is, at the least fair knot, according to the local fairness criterion.
- Recalculation of $G_s$ and back to the previous step.

### 4.2. Local fairing iteration

The smoothness of the surface at the knot $(u_k, u_l)$ changes from $C^2$ to $C^3$, $L_{kl} = 0$, that is, the third-order derivative discontinuity disappears and the surface is then as smooth as possible at this knot. The local smoothness measure is zero and therefore

$$\Delta_{uuu}(u_k, v_l) = 0; \quad \Delta_{vvv}(u_k, v_l) = 0 \tag{12}$$

This system of equations is compatible but undetermined, since there are more unknowns than equations. We obviously intend to modify the surface minimally, that is, the distance $\max|s(u, v) - s'(u, v)|$ should be minimal. But this condition is highly non-linear.

In order to linearize the problem, while influencing minimally the surface, we modify the position of just nine points, instead of 21, keeping fixed the others. That is, we improve the local fairness measure of each knot $(u_k, v_l)$ modifying the position of the control points $V_{ij}$, $i = k-3$, $k-2$, $k-1$, $j = l-3$, $l-2$, $l-1$, that affect the most the neighbourhood of the knot.

In this way, the new position of these nine points will be given by the solution of the system of Eq. (12) that at the same time causes the smallest deformation of the original surface, that is, the minimum of the vector function $F(V'_{ij})$

$$F(\mathbf{V}'_{ij}) = \sum_{i=k-3}^{k-1} \sum_{j=l-3}^{l-1} |V_{ij}^{x} - V_{ij}^{\prime x}|^2, \quad \sum_{i=k-3}^{k-1} \sum_{j=l-3}^{l-1} |V_{ij}^{y} - V_{ij}^{\prime y}|^2, \quad \sum_{i=k-3}^{k-1} \sum_{j=l-3}^{l-1} |V_{ij}^{z} - V_{ij}^{\prime z}|^2 \tag{13}$$

where $V_{ij}$ is an original control point and $V'_{ij}$ is a modified one.

This condition replaces the minimal deformation condition and it has the advantage of producing linear Lagrange equations.

In order to solve this problem, we make use of Lagrange multipliers,

$$\boldsymbol{\Theta}\left(\mathbf{V}'_{ij},\lambda,\mu\right) = \mathrm{F}\left(\mathbf{V}'_{ij}\right) + \lambda\cdot|\Delta_{uuu}(\mathrm{u_k},\mathrm{v_l})| + \mu\cdot|\Delta_{\mathrm{vvv}}(\mathrm{u_k},\mathrm{v_l})| \tag{14}$$

by imposing that the new function $\Theta(V'_{ij})$ should meet a minimum.

This furnishes the following system of 11 vector equations,

$$\begin{aligned} &\frac{\partial\Theta}{\partial\mathbf{V}'_{ij}} = 0 \quad \text{(9 equations)} \\ &\frac{\partial\Theta}{\partial\lambda} = 0 \quad \text{(1 equation)} \\ &\frac{\partial\Theta}{\partial\mu} = 0 \quad \text{(1 equation)} \end{aligned} \tag{15}$$

In the case of equally spaced knots, the coefficient matrix of the previous system of equations does not depend on the knot. If the differences $u_k - u_{k-1}$ and $v_l - v_{l-1}$ are unity.

$$A = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 & 16 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 16 & -6 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & -24 & -24 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 16 & -6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & -6 & 16 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 4 & 4 \\ 4 & -6 & 4 & 16 & -24 & 16 & 4 & -6 & 4 & 0 & 0 \\ 4 & 16 & 4 & -6 & -24 & -6 & 4 & 16 & 4 & 0 & 0 \end{pmatrix}$$

$$A\cdot\begin{pmatrix} \mathbf{V}'_{k-3,l-1} \\ \mathbf{V}'_{k-2,l-1} \\ \mathbf{V}'_{k-1,l-1} \\ V'_{k-3,l-2} \\ V'_{k-2,l-2} \\ V'_{k-1,l-2} \\ V'_{k-3,l-3} \\ V'_{k-2,l-3} \\ V'_{k-1,l-3} \\ \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} 2\cdot\mathbf{V}_{k-3,l-1} \\ 2\cdot\mathbf{V}_{k-2,l-1} \\ 2\cdot\mathbf{V}_{k-1,l-1} \\ 2\cdot\mathbf{V}_{k-3,l-2} \\ 2\cdot\mathbf{V}_{k-2,l-2} \\ 2\cdot\mathbf{V}_{k-1,l-2} \\ 2\cdot\mathbf{V}_{k-3,l-3} \\ 2\cdot\mathbf{V}_{k-2,l-3} \\ 2\cdot\mathbf{V}_{k-1,l-3} \\ c_1 \\ c_2 \end{pmatrix}$$

$$\begin{aligned} c_1 = (&\mathbf{V}_{k-4,l-3} + 4\cdot\mathbf{V}_{k-4,l-2} + \mathbf{V}_{k-4,l-1} + \mathbf{V}_{k,l-3} + 4\cdot\mathbf{V}_{k,l-2} \\ &+ \mathbf{V}_{k,l-1}) \end{aligned}$$

$$\begin{aligned} c_2 = (&\mathbf{V}_{k-3,l-4} + 4\cdot\mathbf{V}_{k-2,l-4} + \mathbf{V}_{k-1,l-4} + \mathbf{V}_{k-3,l} + 4\cdot\mathbf{V}_{k-2,l} \\ &+ \mathbf{V}_{k-1,l}) \end{aligned}$$

Once the inverse of the coefficient matrix $A$ is known, the iterations of the fairing process are easy to compute, since they are just a multiplication of the matrix by the independent term vector. $V'_{ij}$ stand for the new position of the control points $V_{ij}$ after a fairing iteration.

### *4.3. Final comments to the fairing process*

This fairing process will not be generally necessary when the surface has been modelled with a low number of control points, but also with low precision to fit the hull data, since the approximation procedure has already erased small irregularities that may have arisen in the ship offsets. In Fig. 12, the number of control points may be seen, 7×6, and also the longitudinal distribution of curvature along isoparametric curves, a usual technique for checking the hull fairness.

The problem arises when very high precision is required and a large number of control points are therefore used, even as much as data points. In this case, approximation becomes interpolation. In such cases the surface reproduces precisely the ship offsets, but the feared bumps may appear. We may easily detect them plotting the gaussian curvature of the surface or the curvature of isoparametric curves in any CAD application by importing to an IGES file the data that have been obtained in the previous step.

In Fig. 13, a control net of 7×10 points has been used for fitting ship offsets following the method described in this work. A greater number of control points has been used to obtain a better fitting in the stem region. The control net as well as the longitudinal curvature of the isoparametric curves of the surface may be seen.

In these cases we use the fairing algorithm that we have previously described, which is local, that is, it just modifies the surface where it is needed to be smoothed, altering the information of a few control points and leaving the rest unaltered.

However, if we do not want to alter the shape of the ship hull surface at the deck or at the centreline, or the tangent directions on such edges, we must not allow the fairing algorithm to modify the position of every control point. These parameters affects ship seakeeping.

In order to avoid it, we fix the position of the last and of the first two longitudinal rows of control points. We thereby prevent the algorithm from changing them and therefore the centreline and the bow and stern profiles, as well as the deadrise angle or starting angle of the stations at the centreline, remain unaltered.

In this way, we avoid deforming too much the original hull surface. In order to achieve this, we just restrict the application of the fairing algorithm to a set $I_0$ of inner knots, subset of the

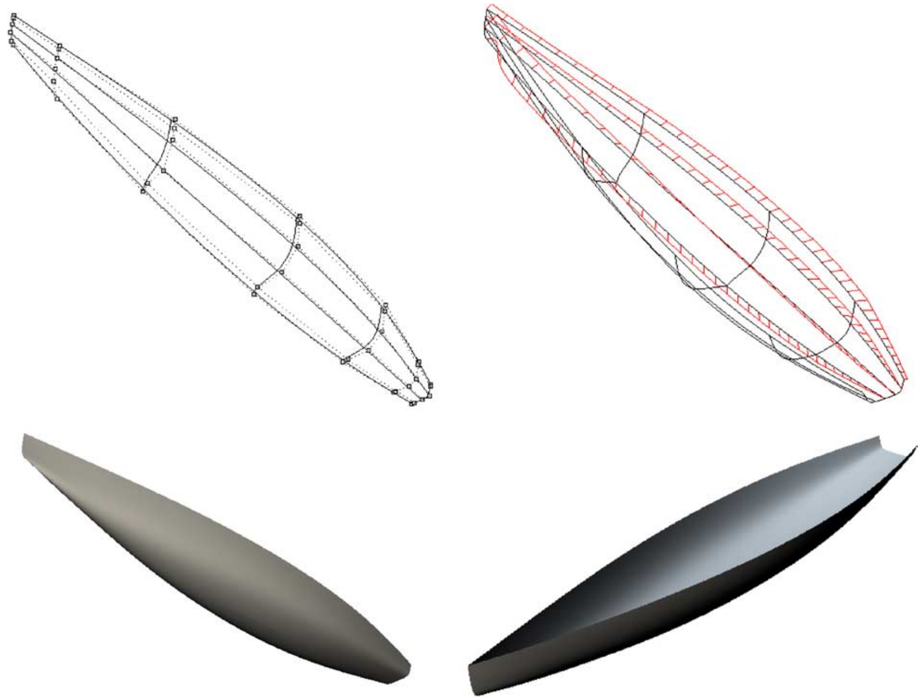


Fig. 12. Unfaired net.

original set $I$, where the last and the first rows of inner knots have been excluded. That is, the knots that are included in the algorithm and therefore may be faired are $(u_k, v_l)$ such that $k$, $l$ are in $I' = \{(4, 4); (4, 5); \ldots; (m-2, n-2)\}$.

In Fig. 14, the fairing effects may be seen on the hull of Fig. 13. The curvature of isoparametric curves has improved if we compare them with the ones that have not been faired.

A last thing to be considered is the required number of iterations of the fairing algorithm. There is no a priori criterion to choose it, since it depends on the considered hull. It is not convenient to use a too large number so that the original surface does not change too much, but it cannot be so low that the surface is not enough faired. A quick view of the curvature of the isoparametric lines is a good indicator.

## 5. Discussion

In this paper, we have described a full process for automatically designing faired hull forms directly from the data points in ship offsets. The method relies on a nice fitting of data points of the stations with an original selection of the parametrization, increasing as much as possible the precision of the approximating cubic spline curves. The origin of the method is the traditional design with physical splines.

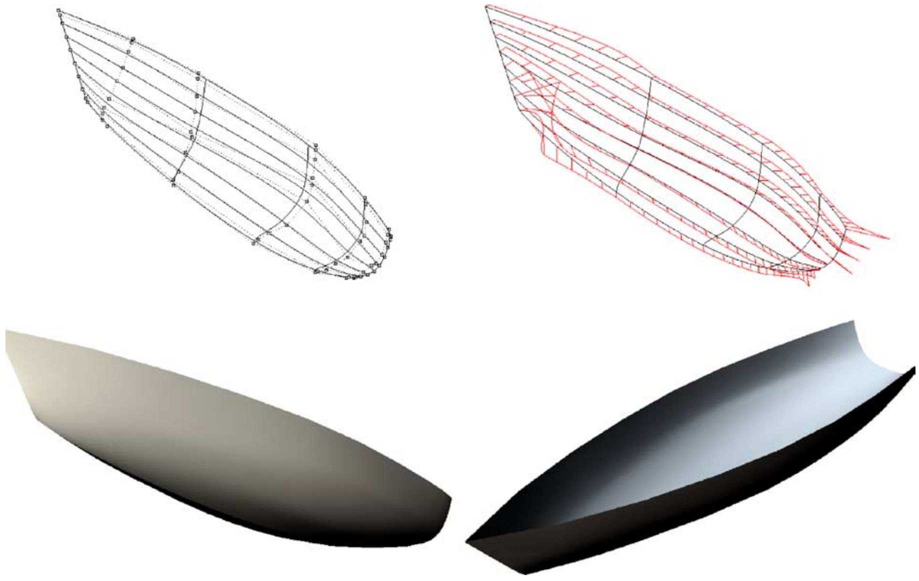


Fig. 13. Unfaired net.

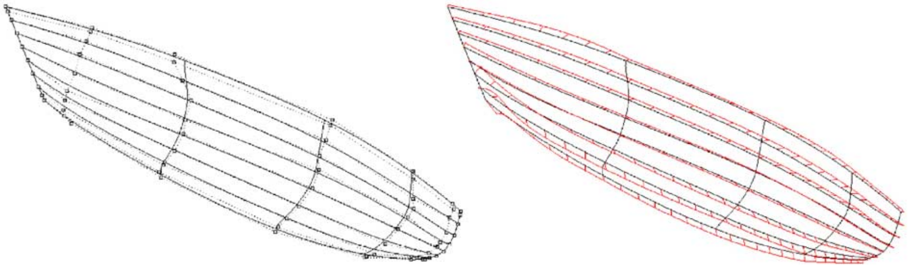


Fig. 14. Control points net with few control points.

To this aim, an iterative algorithm has been devised for obtaining the values of the optimised parametrization for the spline curves that approximate the hull data points. Other parts of the process, the approximation scheme or the fairing algorithm are standard, although the fairing has been adapted in order to maintain certain ship characteristics. The method provides accurate results for most hull forms.

The method, however may be improved in several ways. For instance, other fairing algorithms may be tried such us directional methods [18], because for an elongated shape as the ship hull, it differs much between the transversal and longitudinal directions. This could be another line of research for further improving the method.

## Acknowledgements

Mr Juán Antonio Clemente from Navantia.

## References

[1] Rogers D, Adams JA. Mathematical elements for computer graphics. New York, NY: McGraw Hill; 1990. p. 250–2.

[2] Saunders H. Hydrodynamics in ship design. SNAME; 1957 [Chapter 24].

[3] de Boor C. A practical guide to splines. Berlin: Springer; 1972.

[4] Farin G. Curves and surfaces for CAGD. 5th ed. San Francisco, CA: Morgan Kaufmann; 2001.

[5] Hoschek J, Lasser D. Fundamentals of CAGD, AK Peters; 1993.

[6] Piegl L, Tiller W. The NURBS book. Berlin: Springer; 1997.

[7] Lee E. Choosing nodes in parametric curve interpolation. Comput Aided Des 1989;21:363–70.

[8] Eck M, Hadenfeld J. Knot removal for B-spline curves. Comput Aided Geom Des 1995;12:259–82.

[9] Farin G, Rein G, Sapidis N, Worsey AJ. Fairing cubic B-spline curves. Comput Aided Geom Des 1987;4:91–104.

[10] Sapidis N, Farin G. Automatic fairing algorithm for B-spline curves. Comput Aided Des 1990;22:121–9.

[11] Kjellander JA. Smoothing of bicubic parametric surfaces. Comput Aided Des 1983;15:289–93.

[12] Hahmann S. Shape improvement of surfaces. Comput Suppl 1998;13: 135–52.

[13] Brunet P. Increasing the smoothness of bicubic spline surfaces. Comput Aided Geom Des 1985;2:157–64.

[14] Hahmann S, Konz S. Knot-removal surface fairing using search strategies. Comput Aided Geom Des 1998.

[15] Nowacki H, Reese R. Design and fairing of ship surfaces. In: Barnhill RE, editor. Surfaces in CAGD; 1983. p. 121–34.

[16] Moreton H, Séquin C. Surface design with minimum energy networks. Proceedings of the symposium on solid modelling foundations and CAD/CAM applications, Austin, Texas, 5–7 June; 1991. p. 291–301.

[17] Welch W, Witkin A. Variational surface modelling. Proceedings of SIGGRAPH, Chicago; 1992. p. 26–31.

[18] Feng Jin. Directional surface fairing of an elongated shapes. Proceedings of the FAIRSHAPE conference, 14–17 September, Kleinmachnow, Germany; 1997. p. 163–78.

[19] Hoschek J. Intrinsic parametrization for approximation. Comput Aided Geom Des 1998;5:27–31.

[20] Saux E, Daniel M. An improved Hoschek intrinsic parametrization. Comput Aided Geom Des 2003;20:513–21.

**Dr Francisco L. Pérez Arribas** is an Associate Professor in the Naval Architecture and Marine Engineering School of Madrid (ETSIN), UPM. He received BS and MS degrees in Naval Architecture from the ETSIN Towing Tank in CAD and CASHD. His current research interests include ship hull modelling, parametric ship design and geometric modelling.

**José Antonio Suárez** is a PhD student at the ETSIN. He received BS and MS degrees in Naval Architecture from the Universidad Politécnica de Madrid. His research activities include parametric ship design and hydrodynamics.

**Dr Leonardo Fernández-Jambrina** is Professor of Applied Maths at the Universidad Politécnica de Madrid, Spain, since 2000. He has worked previously as researcher and assistant professor at the Universidad Complutense de Madrid, where he received his Bachelor of Science degrees both on Mathematics and Physics and his Ph.D. on Physics. His current research interests include computer-aided design and geometric modeling with applications to naval architecture.