

# An industrial ship system for the flat development of undevelopable surfaces: algorithm and implementation

E. M. Soto<sup>1</sup>, X. A. Leiceaga<sup>2</sup> & S. García<sup>2</sup>

<sup>1</sup>*AZTECA Consulting de Ingeniería, Vigo, Spain*

<sup>2</sup>*Esc. Tec. Sup. de Ingenieros Industriales, Univ. de Vigo, Vigo, Spain*

## Abstract

This article presents an implementation of a new algorithm for flat development of undevelopable surfaces, which are characteristic of the plates used for construction of ship hull, and designed as B-spline patches, in an Industrial Ship System. The displayed method is based on the structured triangular composition of the surface, for later acquisition of flat development, thereby making differences between the distances of the 3D surface and the flat development minimum. A complete tool integrated in the Industrial Ship System has been obtained, which provides precision and rapidity, and it allows us to obtain all the necessary information for the later production for cutting and forming

*Keywords: undevelopable surface, B-spline, mesh, design patches, hull strake, production.*

## 1 Introduction

This article presents a complete tool, integrated in Naval Engineering System [5], which does the flat development of the complex surfaces that form the steel plates of the ship's hull.

At the moment, in Spanish shipyards the production of curved steel plates is carried out by combined processes [1] mechanical forming and thermo-mechanical forming. First the plate goes through rolls and then, in order to obtain the exact curvature, it is subjected to combined heat and cold processes, this method is known as "forming by line heating". [2].



With the developed application, from the 3D plate model, designed in a generic program, MicroStation V8 – Bentley Systems, its flat development and the necessary lines for its later manufacture in the shipyard are obtained [1].

So it has been necessary:

- To create mesh of the surface. Due to the difficulty in obtaining uniform mesh with the own tools of the selected program.
- To develop the algorithm to obtain the equivalent flat surface.
- To identify the reference lines for manufacture and verification

The programming work has been carried out in M.D.L. (MicroStation Development Language) native language of MicroStation similar to C.

## 2 Resolution of mesh

The ship hull is designed using a set of B-Spline surfaces (Figure 1). The constructive patches do not coincide with these B-Spline surfaces.

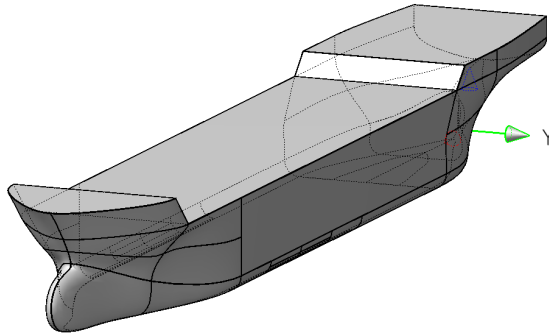


Figure 1: Set of B-Splines surfaces that compose hull ship.

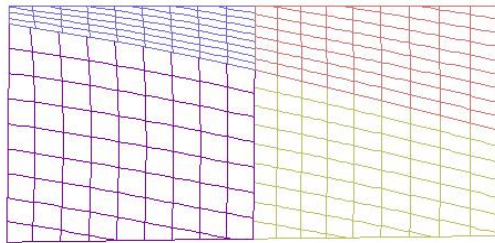


Figure 2: Constructive patch made up of 4 surfaces of design.

The first difficulty is to obtain uniform mesh in the constructive patches composed of several B-Spline surfaces as shown in Figure 2. Therefore we decided to program a mesh application of all the cases and this tool maintains the possibility of visualizing it, thus being able to select the optimal mesh for each plate.

The algorithm mesh is made up of the following steps:

Step 1: Make an approximated triangulation of the surface. Search for the edges (condition: without neighbouring edges).

Step 2: Obtain the best direction of projection to avoid loss of information according to different criteria. (Figure 3.)

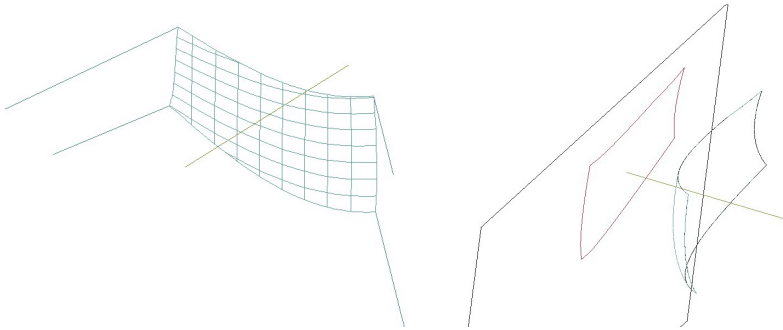


Figure 3: Direction of projection.

Step 3: Obtain the best approach of the edges of the constructive patch like B. Spline.

Step 4: Make a flat surface on which to project the edges obtained in the previous step. This flat surface will be perpendicular to the direction of selected projection.

Step 5: Make the projection of the curves obtained as limits of the patch on the flat surface. Thus, we obtain a flat Spline surface, with which we will work in a simpler way.

Step 6: Extract the isoparametric curves of this flat surface. We obtain the surface mesh as intersections of these curves. The intersection points are the regular and square mesh, optimal for later processing.

Step 7: Projection of these points onto the original surface. If the projection does not intersect the original surface, we search in the triangulation of the original surface.

This is the solution found, results are good and it is the only one which has not produced any problems with surfaces composed of several patches of design.

### 3 Development algorithm

The proposed method is based on the non-structured triangular composition of the surface, to later obtain its flat development so that we can minimize [3] the difference between the distance in the flat development and the same distance on the original surface.

The development of the surface will be done by minimization of an “application error” (see below). This is the difference between the distances of the real surface and the same distance in the developed surface between neighboring points, by application of local isometrics between the real surface and the mesh of developed surface.

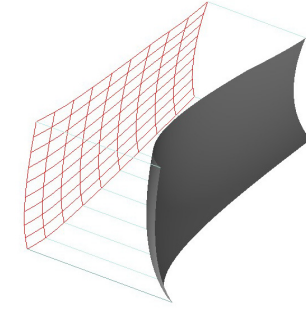


Figure 4: Mesh is projected on the original surface.

$$E = \sum_{k=1}^l \left[ d_k^{3D} - d_k^{2D} (v_{1x}, v_{1y}, \dots, v_{ix}, v_{ix}, \dots, v_{pqx}, v_{pqy}) \right]^2 \tag{1}$$

where:

- p: number of points distributed in parametric direction  $u$ .
- q: number of points distributed in parametric direction  $v$ .
- $v_{ix}, v_{iy}$ : variables of the unknown function, that correspond to coordinates  $x$  and  $y$  in development surface  $D$ , corresponding to  $p \cdot q$  points of  $S$
- $d_k^{3D}$ : distance in  $S$  (3D) between neighboring points.
- $d_k^{2D}$ : distance in  $D$  (2D) between neighboring points in  $S$ .
- $l = 4 \cdot p \cdot q - 3 \cdot (p+q) + 2$ : number of total segments that connect neighboring points.
- $i = 1, 2, \dots, p \cdot q$ : number of total nodes of the parametric net.

The concept of development is to look for the minimum of this function [3], that is to say, to look for a dimension vector  $p \cdot q, v^*$ , so that:

$$E(v^*) = \min E(v) = \min \sum_{k=1}^l \left[ d_k^{3D} - d_k^{2D} \right]^2 \tag{2}$$

In this way, a triangular flat mesh is obtained which is approximate to the real one, where the difference between corresponding distances is minimum, between the 3D surface and the plane.

Also we must consider in the minimization process the conditions of contour in order to look for a greater efficiency in the result.

### 3.1 Step 1: Definition of the parametric space of the mesh

The first step of the algorithm consists of arrangement of the data maintaining the defined parametric space according to Figure 5.

### 3.2 Step 2: Division in two meshes for the application of the algorithm

The procedure followed in this algorithm is to construct *two new meshes*: original (Mesh A) and another turned by  $180^\circ$  with respect to the first (Mesh B). That is to say, one which begins the development from the node (0,0) and the

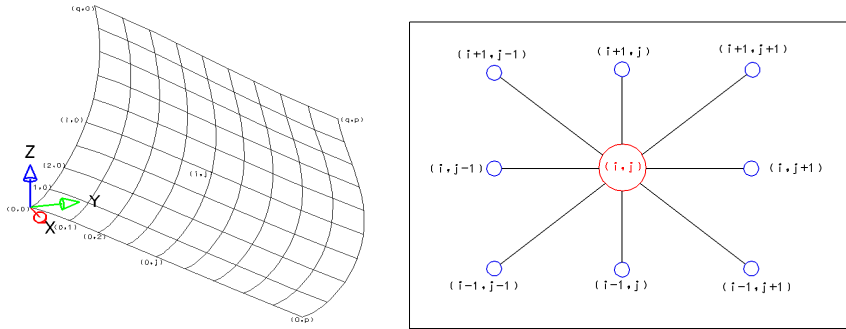


Figure 5: Definition of the parametric space and distances of a point (i, j).

other, applying to the function of rotation of angle 180°, from the node (p,q) (the opposite node).

We calculate the 180° turned mesh by copying the coordinates in the opposite sense of the numeration, we begin by the coordinates of the node (p,q) until arriving at (0,0)

Thus, we obtain two meshes with the same coordinates but ordered in two different ways. It is possible to apply the same functions to both meshes, and thus we obtain two developments from opposite ends of the same surface.

### 3.3 Step 3: Previous calculations

First we will carry out the calculations of the distances and the angles of verification: edges, diagonals, perimeter and patch area; which inform us how good the approach obtained is.

The distances in the 3D surface are measured according to the formula:

$$d_{(i,j)(i,j+1)} = \sqrt{(x_{i,j} - x_{i,j+1})^2 + (y_{i,j} - y_{i,j+1})^2 + (z_{i,j} - z_{i,j+1})^2} \quad (3)$$

All the distances represented for a generic point (i,j) in Fig. 5 must be considered: same row, same column, right diagonal, left diagonal.

There are singular nodes, the nodes of edges and vertices, in which there are not all the distances. These values are eliminated.

In order to obtain the angles, sines and cosines, which are necessary for the first points of the approach, we used the theorem of the cosine with the distances already defined. First, we calculate cosine – equation (4) – and from them, the angle and the sine

$$\cos \gamma_2 = \frac{d_{(0,0):(0,1)}^2 + d_{(0,0):(1,0)}^2 - d_{(1,0):(0,1)}^2}{2 \cdot d_{(0,0):(0,1)} \cdot d_{(0,0):(1,0)}} \quad (4)$$

### 3.4 Step 4: Approach calculations

Thus, we have all the necessary values to begin the calculation of the approach; we will apply the same calculations to two meshes, Mesh A and Mesh B.



In order to obtain the starting points with which to diminish the function error we calculate the average of the data obtained to fit to the distances in rows of the original surface ( $v_{(i,j)}^1$ ), and to the distance in columns ( $v_{(i,j)}^2$ ) for each of the meshes, that until now stays separated.

$$(v_{(i,j)x}^0, v_{(i,j)y}^0) = \left( \frac{v_{(i,j)x}^1, v_{(i,j)x}^2}{2}, \frac{v_{(i,j)y}^1, v_{(i,j)y}^2}{2} \right) \quad (5)$$

where:  $i = 0, 1, 2, \dots, p$ ;  $j = 0, 1, 2, \dots, q$ . Crossing all the points of the mesh.

Applying the function that has been programmed from this algorithm, two meshes 2D (Mesh A and Mesh B) are obtained, which are defined by  $(p+1) \cdot (q+1)$  - initial vector ( $v(i, j) x_0, v(i, j) y_0$ ), that is to say, the coordinates ( $x$ , and) of  $(p+1) \cdot (q+1)$  nodes corresponding to the nodes of the 3D mesh.

### 3.5 Step 5: Merge two meshes

In order to be able to merge these meshes we must, in the first place, undo the rotation of  $180^\circ$  applied previously to Mesh B.

Once we have the two meshes development with the same order of nodes, we merge those using coefficients by rows and columns, so that the values of the coordinates of the nodes that agreed with the axes of the development mesh will have more weight, because these have a smaller error. In Mesh A is the row 0 and the column 0, whereas in Mesh B is row  $p$  and column  $q$ .

The coefficients are determined following the formulas to Mesh A, whereas the coefficients of Mesh B will be 1 minus the Mesh A coefficients:

$$Coef.colum = 1 - \left( \frac{nf \cdot p}{N - p} - \frac{1}{p - 1} \right) \quad (6)$$

$$Coef.fila = 1 - \left( \frac{nc \cdot q}{N - q} - \frac{1}{q - 1} \right) \quad (7)$$

where:  $nc$  = number of the column of the node;  $nf$  = number of the row of the node;  $p$  = number of columns;  $q$  = number of rows;  $N$  = number of total nodes

Applying the weighted average of the two meshes from the equations:

$$Coef.ColumA \cdot x_1 + Coef.ColumB \cdot x_2 = x \quad (8)$$

$$Coef.FilaA \cdot y_1 + Coef.FilaB \cdot y_2 = y \quad (9)$$

We obtain the coordinates ( $x$ ,  $y$ ) of 2D developed meshes. In addition we calculate the edges, diagonals and area (with the same function applied to 3D mesh) we compared them with the data that we obtained in 3D mesh to give us an idea of the error made in the development.

### 3.6 Step 6: Relocate

This step consists of fitting the mesh to the 3D distances of the edges, diagonals and perimeter.

For this, we first measured the necessary distances to compare with those of 3D mesh, later, we made the difference between the 2D and 3D distances, we moved the coordinates in  $x$  and  $y$  accordingly, from the center of the mesh.



First we move the lines individually and later we correct the edges so that the mesh is maintained as uniform as possible. Since there is displacement we must re-compute the measurements between nodes of the 2D mesh. We must do it whenever there is displacement of nodes.

### 3.7 Step 7: Calculation of the error (E)

The error is defined in equation (1) as the square of the differences between the 3D distances and the same in the 2D mesh, according to the four specified directions (horizontal, vertical and the two diagonals)

### 3.8 Step 8: Iterations

Iteration process: process to diminish the function error [5], applying the following calculations:

1. Calculation of the factor, from the calculated errors

$$\left( \frac{2 \cdot E}{d_{(i,j)-(a,b)}} \right) \tag{10}$$

2. Calculation of the difference of the coordinates between the iteration and previous one (being it\_a and it\_b consecutive iterations)

$$(v_{(i,j)x}^{it-a} - v_{(i,j)x}^{it-b}) \tag{11}$$

3. Calculation of the divergence of coordinates, that is the product of both previous iterations.

$$\left( \frac{2 \cdot E}{d_{(i,j)-(a,b)}} \right) \cdot (v_{(i,j)x}^{it-a} - v_{(i,j)x}^{it-b}) \tag{12}$$

4. And the total divergence is the sum of the divergences of the coordinates which affect the neighbouring nodes, whilst still considering the signs.
5. Calculation of the 2D mesh with the previous mesh and the divergence multiplied by a coefficient (by defect 0.1)

$$\nabla E(v) = \sum_i \frac{\partial E}{\partial v_i} \cdot v_i \tag{13}$$

After finding the mesh of the iteration, we do all the calculations again to compare the results and to prepare the necessary values for the next iteration.

- Calculate the distances
- Calculate the error (difference between 3D and 2D)
- Calculate the edges, diagonals and angles; and calculate the difference with the real valour.
- Once all this is calculated, we include a second process that consists of relocating the nodes from the centre of meshes, to improve the approach.
- Calculate lines and relocate
- Calculate frames and relocate



- Calculate the distances 2D
- Relocate the edges
- Distances in the 2D mesh
- Calculate frames
- Calculate lines
- Calculate error
- Measurement edges and diagonals
- Error of the distances to compare them finally with those which we must reach.

A maximum value has been taken from the error to finish the iterations; when the program arrives at that error, the algorithm ends. We also implemented the possibility of introducing a number of iterations to verify the convergence of the algorithm.

## 4 Result: a functional tool

The result is a functional tool for the development of non-developable surfaces implemented in the generic program MicroStation.

We analyse its operation with a block of the Fore Peak, (Figure 6), where the curvatures of hull are more complex. The patches to develop for this block are those of (Figure 7), where they are separated to observe the different forms and curvatures from the same.

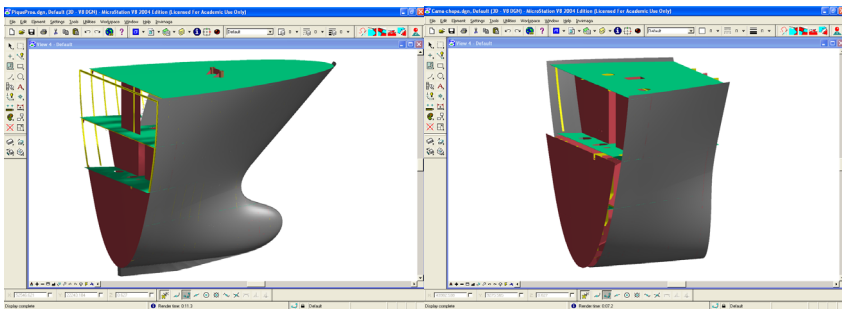


Figure 6.

We apply the programmed tool: “Develop patches” that carry out the explained algorithm. First we select the surface, in our case the patch 02, like we see in fig. 7. A dialogue window appears, where we input values of the different variables for the development.

### 4.1 Selection of parameters

We select, in the first part of the dialogue window, the mesh density, the tolerances of mesh, and the marks of the final developed surface.



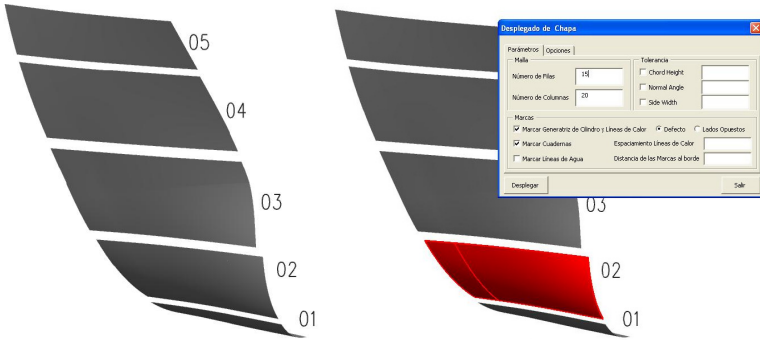


Figure 7.

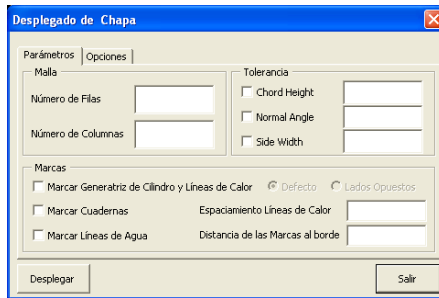


Figure 8: Selections of parameter to develop the surface.

With respect to the mesh we must indicate: number of rows (p) and number of columns (q).

With respect to the error or tolerance of mesh: chord height, normal angle to the surfaces, side width maximum. These variables are of the MicroStation mesh tool, they will have values by defect and it is not necessary to cover them.

Regarding the production marks, that we want to appear in the developed surface, we have the variables: to mark guide lines to roll; to mark Frames; to mark Water Lines; distance between Lines of Heating; Distance from the mark to the edge. These marks are obtained by connection with the System Data Base, where the coordinates are stored.

#### 4.2 Exit Options.

With respect to the second part of the window, we selected the exit options (fig. 9) that can be grouped in:

Options to visualise of 2D Mesh solution. Show in the file on which it works or in another file.

Options to visualize of 3D mesh of the surface to develop. Show in the file on which it Works, or select another file.

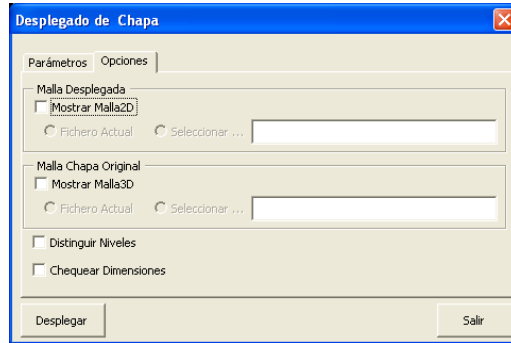


Figure 9: Exit options.

Options to verify the exit data: developed 2D surface and marks. Write each set of elements in a level, to show or hide the data according to want Obtain comparative table with the dimensions obtained in both meshes.

### 4.3 Developed 2D mesh

After we give parameters and exit options, we push the developed button and obtain developed mesh like B-Spline Surface, as shown in the images of fig. 10. The left image is mesh of the 2D surface and the right figure shows lines of marked, frames and guidelines to roll in its corresponding levels.

In fig. 11 we see the comparative tables between the measurements of 2D and 3D surfaces for the verification of the approach. In addition, these distances are used also for later weld fillet calculations, times of the cutting machine, among others. All these data are exportable as TXT file for their study or later calculations.

## 5 Conclusions

We emphasise the following conclusions among others:

We obtain optimal meshes for each surface. We obtain meshes that better adjust to each surface. In this respect, the tool allows the execution and visualisation of 3D meshes for the election of what we considered next to the real surface, and thus the optimal one to obtain the flat development.

The tool is fast. It is important to emphasise the rapidity of the tool, noticing that although we were programming graphical methods, used at present [4], these would be slower.

Definition of finished tool. As the paper has shown, a tool implemented in System Integral of Engineering and Design has been developed. It makes sufficient accuracy in the flat development of the non-developable surfaces that compose the ship hull, in addition it gives all the necessary information for its production; positioning, cuts and forming.

Unique 3D model design. We emphasize too that the developed planes are directly generated from the three-dimensional patches of the model of the hull. A

unique model of design is used for calculations, analysis and production; thereby preventing the possible incoherences that take place at present due to the existence of several models.

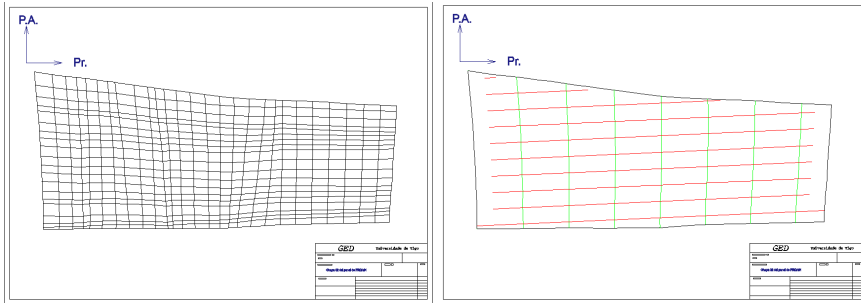


Figure 10: Left: develop surface mesh. Right: lines and marks.



Figure 11: Report of verification of distances.

## References

- [1] Ship design and construction. The Society of Naval Architects and Marine Engineers, New York, 2003.
- [2] Aplicación técnica de calor para conformado. Astilleros Españoles, S.A. November 1990
- [3] William H. Press. *Numerical Recipes*. Cambridge 1992-1996.
- [4] Enrique Pardo Trazado de líneas y desarrollos del buque. Editorial Gustavo Gili, S.A. Barcelona 1984.
- [5] Philipe E. Gill, Walter Murray, Margaret H. Wright, *Practical Optimization*. Academic Press, 1981
- [6] X. Leiceaga, F.G. Zapatero, M. Rodríguez and J. Prieto, *Proyectación aseque de buques. El modelo 3D al servicio de la ingeniería naval*. Revista Ingeniería Naval, Article Technical, October 2003.
- [7] Henrik Bisgaard Clausen. *Plate Forming by Line Heating*. Technical University of Denmark, KGS Lyngby. 2000

