

## A Survey: application of geometric modeling techniques to ship modeling and design

Kwang Hee Ko

*Gwangju Institute of Science and Technology, Gwangju, Korea*

**ABSTRACT:** *In this study, geometric modeling techniques and their application to ship modeling and design are presented. Traditionally the hull shape is defined by using curves called the lines and various necessary computations are performed based on the discrete points obtained from the lines. However, some applications find difficulty in using the lines such as seakeeping analysis, which requires the computation of wetted part that is changing dynamically over time. To overcome such a problem and increase accuracy and efficiency in computation, two essential geometric modeling techniques, surface modeling and surface-to-surface intersection, are introduced and their application to ship modeling and analysis including hydrostatic computation, slamming and seakeeping analyses is presented.*

**KEY WORDS:** Surface-to-surface intersection; Geometric modeling; Ship modeling; NURBS; CAD.

### INTRODUCTION

Traditionally the hull form of a ship is defined by curves in three planes, each of which is called the body, sheer, and half breadth plans. A set of these drawings is called the lines, which is the starting point of ship manufacturing (See Fig. 1).

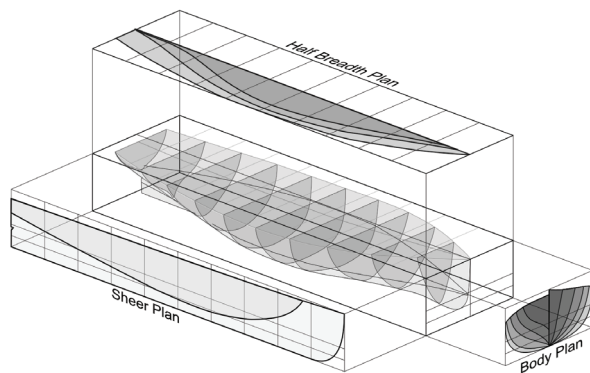


Fig. 1 The illustration of the lines (Tosaka, 2001).

The lines consist of curves and each curve defines the shape of a ship. From the body plan, the shape at each station or frame is defined. The shape of the hull at each water line is given in the half breadth plan and the shape of the hull at fixed distance from the centerline, also called the buttock lines, is defined in the sheer plan. All the calculation and design work is performed with the lines.

The intersection points of the curves are computed from the lines, which are called the offset data of a hull shape. The offset data are arranged in a table, called an offset table, which is the usual way to deliver the shape of the hull because curves drawn in the lines cannot be extracted in an exact manner. Since the offset points are the discrete points obtained from the curves, they inevitably lose shape information during the offset generation. This means that it might be possible, if it does not happen frequently, to obtain different lines even from the same offset points.

To overcome this problem, various CAD systems such as Tribon, which support lines creation using various mathematical curve definitions, were used. These systems can deliver curves directly without generating offset points to the subsequent processes. The use of such systems is considered as an innovation in the ship design. However, the way of defining the hull shape using curves has not been changed and the process of the lines generation, that was done manually, has been partly automated using the systems.

The main issue in defining a hull shape is that the hull shape can be exactly given only by using surfaces, not curves, which only provide shape information at specific locations. The regions that curves are not defined need to be estimated from the given curves. This means that the use of curves has the limitation of representing the hull shape accurately. To address such a problem, a set of new CAD systems which support modeling of a ship using surfaces are introduced with the help of software and hardware development. The CAD systems such as CATIA can generate surface models for the hull shape and process them for calculation, simulation, detailed design and manufacturing. In fact, they have evolved to provide a solution which can take care of the entire shipbuilding.

Surface modeling of a shape is a difficult process requiring a wide range of geometric modeling techniques. However, after obtaining surface models, a variety of computation and design steps can be done efficiently and accurately since shape information at any point on the surface can be obtained without an additional estimation step from the curves. For example, one can extract the shape of a ship at any position directly. In addition hydrostatic and dynamic computation can be performed efficiently. Such computation is mostly carried out using meshes that approximately define the hull shape and the meshes can be more efficiently obtained from the surface models than curve models.

Among the various geometric processes for surface models, computation of surface-to-surface intersection (SSI) is the most difficult but an essential job and finds many applications in ship design. For example, extracting lines from the surface models can be easily done by intersecting the hull shape with various planes and computing the intersection curves. Similarly, the draft line of a ship at a certain free surface can be obtained by computing the intersection of the hull and the free surface.

In this paper, a brief review of two techniques, surface modeling and SSI computation, is presented to give an insight on the importance of the techniques in the ship design phase and a couple of examples are introduced which can be done efficiently by using the surface model of a ship hull and SSI computation with its theory. Application of SSI computation to hydrostatic calculation, slamming and seakeeping analyses is introduced. The hydrostatic computation requires the exact shape information and the wetted part of the hull floating on still water. Slamming needs the body shape at an arbitrary position. Seakeeping receives meshes of the wetted part of the ship hull given various free surface conditions.

The paper is structured as follows. Geometric operations for surface modeling and SSI computation are introduced and how such operations are used in the computations mentioned above. Examples are followed and this paper concludes with future work.

## GEOMETRIC MODELING

In this section, the theoretical background of two essential geometric operations: surface modeling and SSI computation are introduced.

### Surface Modeling

A surface can be represented using implicit, explicit and parametric form. Among them, the parametric representation is mainly used in the surface modeling since it provides computational advantages and geometrically intuitive ways for handling the shape. In particular one typical parametric representation, NURBS (Non-Uniform Rational B-Spline), is the most popular. It is also accepted as the industrial standard in representing surfaces. A NURBS surface is given as follows (Patrikalakis and Maekawa, 2001):

$$\mathbf{r}(u, v) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} \mathbf{p}_{ij} N_{i,m}(u) N_{j,n}(v). \quad (1)$$

Here,  $\mathbf{p}_{ij}$  is the control points,  $N_{i,m}(u)$  and  $N_{j,n}(v)$  the B-spline basis functions of order  $m$  and  $n$ ,  $n_u$  and  $n_v$  the number of control points. This surface is defined over the knot vectors  $\mathbf{T}_u$  and  $\mathbf{T}_v$ .

There exist two ways to generate a surface of the desired shape. First, one surface having a simple shape is given. Then the control points are adjusted to change the surface shape until the desired shape is obtained. The second approach is to find a surface which approximates or interpolates a set of points or a family of curves that containing part of the desired shape. Then this surface is adjusted to become the desired shape. This way of modeling requires estimating parameters for given points or curves and solving a system of equations to determine the control points, which are not an easy process. However, in the general hull form design, the existing shape is taken and changed to satisfy new design requirements. Therefore, the latter method is usually preferred since reference data are available for modeling in ship design.

Suppose that a set of points  $\mathbf{r}_k$  ( $k=1 \dots n_k$ ) are given. These points define the shape of interest. The parameter values for each point  $\mathbf{r}_k$  need to be estimated first. There are several methods for estimating parameters from points and among them the chord length parametrization is mostly used. Assume that the parameter value for  $\mathbf{r}_k$  is given as  $u_k$  and  $v_k$  and knot vectors  $\mathbf{T}_u$  and  $\mathbf{T}_v$  for each parametric direction are provided. Then for each point, Eq. (2) should hold.

$$\mathbf{r}_k = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} \mathbf{p}_{ij} N_{i,m}(u_k) N_{j,n}(v_k). \quad (2)$$

Here  $\mathbf{p}_{ij}$  are the unknowns. Depending on the modeling strategy, a couple of constraints such as enforcing the boundary points, lines, etc. could be considered. Eq. 2 can be easily solved by using the singular value decomposition (Press et al., 1988).

After obtaining the control points, the surface either interpolates the given points if the number of points is equal to the number of control points or approximates them if the number of points is larger than the number of the unknowns. If the latter is the case, then an additional step could be employed to reduce errors in approximation: parameter correction. Parameter correction is a way to adjust parameter values that have been estimated for each point to reduce the approximating error. A method given in Hoschek and Lasser (1993) may be used for this purpose. When points with parametric values are approximated, a generated surface does not pass the given points in most cases. Therefore, there exist errors between the points on the surface at the estimated parameters  $u$  and  $v$  and the given points corresponding to the  $u$  and  $v$ . Suppose that  $\mathbf{r}_i$  is an input point and  $\mathbf{r}(u_i, v_i)$  is the point on the surface using the parameter values  $u_i$  and  $v_i$  assigned to  $\mathbf{r}_i$ . Then the parameter values  $u_i$  and  $v_i$  are revised to reduce the error between  $\mathbf{r}_i$  and  $\mathbf{r}(u_i, v_i)$ . This process is based on the fact that the projection point of  $\mathbf{r}_i$  onto  $\mathbf{r}(u, v)$

yields the minimum error, which is formulated as an iterative process, see Hoschek and Lasser (1993). Optionally, a direct error reduction step can be considered. This step computes the error between the points and the approximated surface. Then the amounts of adjustment for each control point are computed and added to the current control points to reduce error. An example of surface modeling and error reduction is given in Fig. 2. As shown in the figure, after error reduction the error between the points and the surface has decreased.

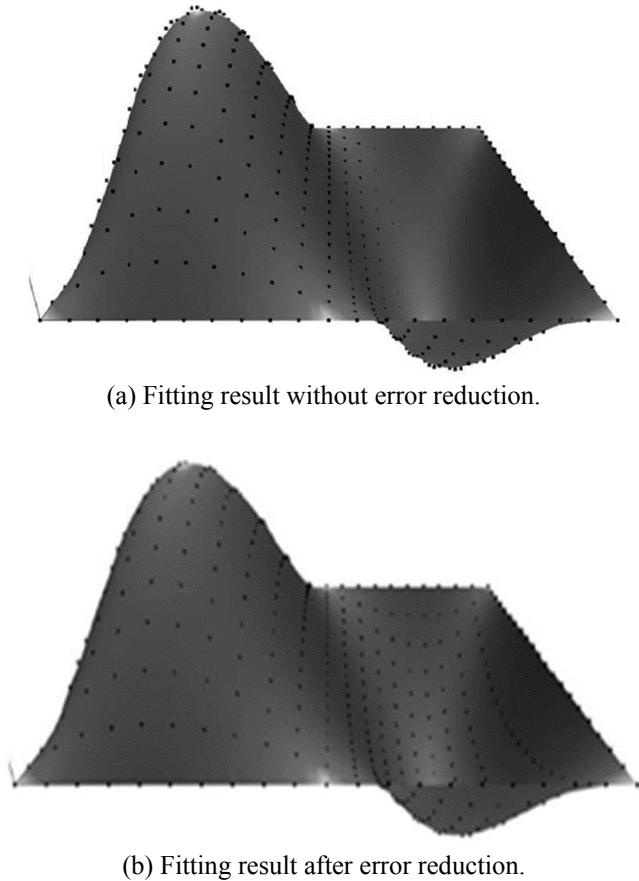


Fig. 2 An example of surface modeling from a set of points.

### Modeling of Hull Shape

Using the surface modeling technique, a ship hull can be defined with surfaces. In general it is not possible to define the entire hull shape with one surface patch. So, the hull is subdivided into several regions in order to make the complicated shape smaller but simpler and each region is approximated using a surface patch. An example of the surface model of a ship is given in Fig. 3. The ship hull in this example is constructed by using five patches of degree 3 in  $u$  and  $v$  directions. The errors before and after the error minimization step are given in Table 1. Here, the error is the maximum distance from the input points to the approximated surface. As shown in the table, the error reduction step minimizes errors between the given points and the approximated surfaces.

Table 1 Error reduction for surface modeling.

Patch No.	Before	Parameter correction	Control point adjustment
1	154.53	8.74	0.14
2	99.12	3.31	0.08
3	106.98	4.49	0.06
4	104.98	3.80	0.07
5	214.09	9.00	0.27

The generated patches are connected with  $C^0$  continuity. This continuity can be achieved by using the common boundary for the two consecutive patches. Higher continuity conditions such as  $C^1$  and  $C^2$  can be considered. However, such continuity conditions are no longer taken into account when the surface patches are discretized into panels or meshes for various numerical computations, which in general require  $C^0$  continuity between consecutive elements.

### Surface-to-Surface Intersection

Consider two parametric surfaces  $\mathbf{r}_1(u, v)$  and  $\mathbf{r}_2(\sigma, t)$  with  $0 \leq u, v, \sigma, t \leq 1$ . If the two surfaces intersect, Eq. 3 should hold at the intersection.

$$E(u, v, \sigma, t) = (\mathbf{r}_1(u, v) - \mathbf{r}_2(\sigma, t)) \cdot (\mathbf{r}_1(u, v) - \mathbf{r}_2(\sigma, t)) = 0 \quad (3)$$

Then, SSI computation is reduced to find the parametric values  $u, v, \sigma$  and  $t$  which satisfy Eq. 3. There exist three major ways for SSI computation: lattice method (Rossignac and Requicha, 1987), subdivision method (Lane and Riesenfeld, 1981) and tracing method (Bajaj et al., 1988; Barnhill and Kersey, 1990).

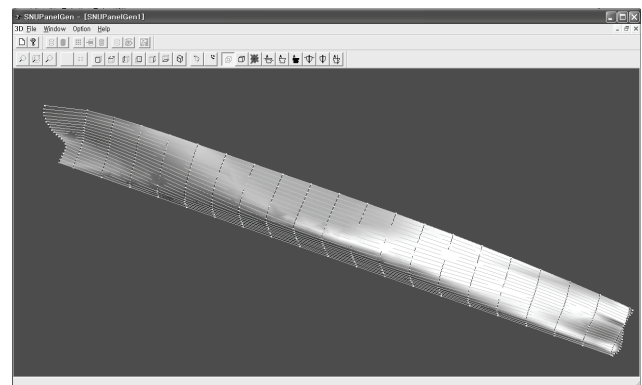


Fig. 3 An example of the surface model of a ship.

### Lattice Method

It transforms surface-to-surface intersection to iso-parametric curve-to-surface intersection. From one of the intersecting surfaces, a set of iso-parametric curves is obtained each of which is taken for computing intersection with the other surface. Then the intersection points are connected to form the intersection between the two surfaces.

This method, however, has a drawback that depending on the resolution of the iso-parametric curves, small intersection features could be undetected.

#### Subdivision Method

It involves recursive subdivision of the surfaces and checks if the subdivided regions from the surfaces contain intersection or not. The subdivision continues while eliminating those regions which do not contain intersection until the size of the subdivided regions becomes less than the user provided tolerance. Figure 4 illustrates how subdivision method works. In this figure the shaded region indicate that they contain intersection.

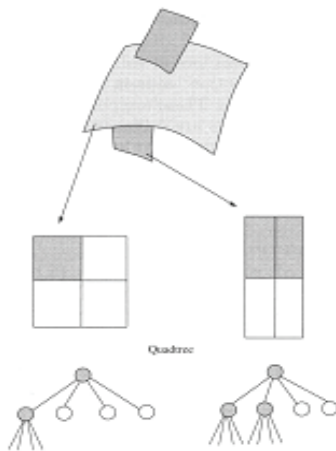


Fig. 4 A schematic of subdivision method (Patrikalakis and Maekawa, 2001).

The mathematical way to determine if a region does not contain intersection is to use interval arithmetic (Moore, 1966) which is applied to Eq. (3). Namely, each subdivided region is represented using interval values. Then using the interval values, Eq. (3) is evaluated in interval arithmetic. If the resulting interval does not contain zero, then it is determined to have no intersection and is discarded. This method can find every intersection feature. However, after the computation, the subdivided regions should be arranged to form intersection and the number of subdivided regions could explode depending on the termination tolerance size and the pattern of intersection (tangential or near tangential intersections), which may lead to increased processing time. Nevertheless, due to its robustness, easy concept and implementation, this method is frequently used in practice.

#### Tracing Method

It generates a sequence of intersection points by solving a system of nonlinear ordinary differential equations that are formulated based on Eq. (3). It requires identifying the starting and ending points of each intersection segment as well as various critical and singular points. This identification step is called the topological configuration. Then, the governing differential equations are solved to trace each intersection curve numerically. This method efficiently computes the intersection. However, the topological

configuration step which sometimes requires a lot of computation should be preceded and for a tangential intersection case, a different formulation needs to be solved, which adds extra complexity to the method. For details, see (Patrikalakis and Maekawa, 2001).

Among the three methods, the subdivision method is employed in the examples in this paper since its implementation is straightforward and the method can efficiently handle complex intersection cases such as the intersection between the hull shape and free surface. The subdivision method is designed to deal with intersections of any type of surfaces from planes to free-form surfaces in one algorithm with no modification. Therefore, it can be applied for various applications easily.

## HYDROSTATIC CALCULATION

Hydrostatic calculation is a fundamental step in ship design. In particular, values from the hydrostatic computation will affect the major performance as well as the safety of the ship hull. Therefore, accurate computation of such values is the most critical. So far computation has been done from the lines. Data points are obtained from the lines and then various numerical integration algorithms such as trapezoidal method, Simpson rule method, etc. are employed for calculating essential values such as displacement, LCB, etc. However, it would be obvious that such computation can be accurately and efficiently performed if the hull shape is given in surfaces. First, the wetted part of the hull shape is computed. This process is done by intersecting the hull shape with planes positioned at the design and ballast drafts. Also, various trim conditions are also considered in the similar manner. Namely, the hull is intersected with a sloped plane. Then the part below the intersection is the wetted one and only that portion is taken for computation.

## SLAMMING COMPUTATION

Slamming is the impact on the hull structure against the water and the estimation of the slamming force is very important for structure design. Given a hull shape, the slamming force can be estimated by several methods: a method using Von Karman theory (Von Karman, 1929), Wagner's method (Wagner, 1932), etc. and most of them require the body profile at an arbitrary position. For this computation, using the lines needs additional work to estimate values if that position is between two frame lines. However, extracting the body shape can be easily done by using the surface model of a hull shape. The hull model is intersected with a vertical plane which is parallel to the body plan and located at the desired position. The intersection curve becomes the body profile. Fig. 5 shows the body plan with five computed profiles which are drawn in thick curves and Fig. 6 presents another view of the same result.



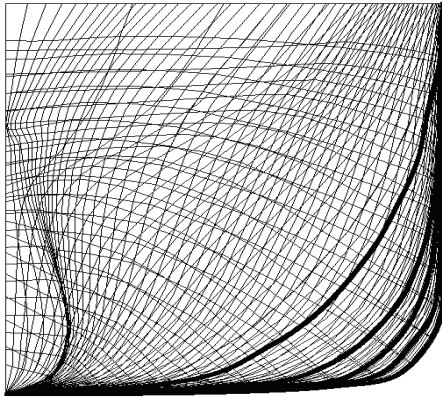


Fig. 5 A body plan showing the profiles at five arbitrary positions (Blue curves: boundary curves of each surface patch. Thick curve: computed profiles).

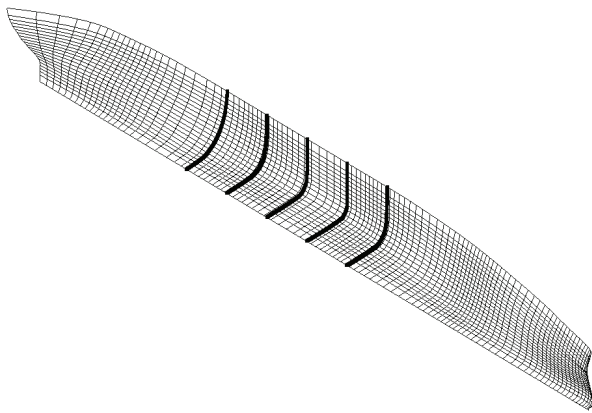


Fig. 6 The computed profiles given as thick curves.

## SEAKEEPING ANALYSIS

Seakeeping analysis is a process of estimating fluid forces acting on the ship hull for the design of the ship structure and the shape. This analysis requires triangular or quadrilateral meshes of the wetted part of the ship, which should be computed efficiently and accurately given the ship motion and wave conditions. For this computation, first, the intersection between the free surface with/without waves and the shape of the hull is computed and the parts below the free surface are discretized into meshes which are then used as input for seakeeping analysis.

Seakeeping analysis can be performed in different ways. Among them, in this work, Rankin panel method is considered. In this analysis method, there exist three types of analysis depending on the degree of nonlinearity (Kim and Kim, 2009): linear, weakly nonlinear and weak-scatterer. The linear analysis needs panels for the wetted part which is obtained when the hull floats on the still water. This condition is easily obtained by intersecting the plane of the still water with the hull shape and taking the part below the intersection. As shown in Fig. 7, the wetted part is computed. Fig. 8 shows the blown-up image of the bow of the ship and

Fig. 9 presents an example of panels of the wetted part including the free surface, which would be given to the linear analysis.

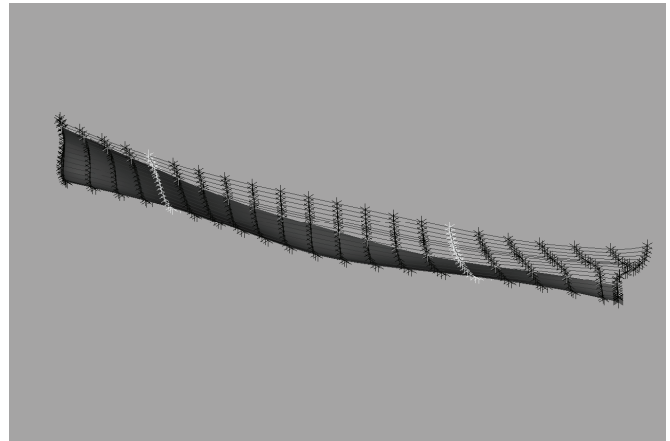


Fig. 7 The wetted part of a ship hull in dark color; The user selected reference stations during hull modeling as yellow curves.

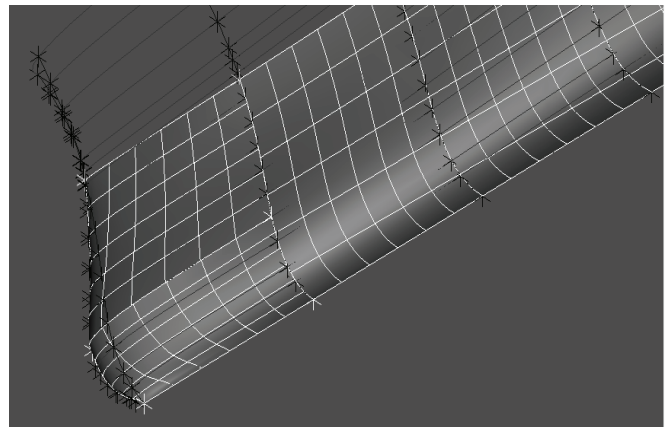


Fig. 8 The magnified bow of the wetted part (Each wetted part represented in surfaces and shown using panels; Waterlines as red curves).

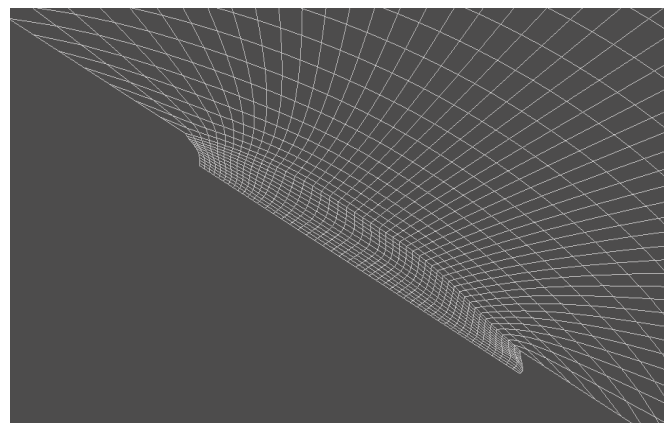


Fig. 9 The generated panels for seakeeping analysis (Ko et al., 2010).

Fig. 10 shows an example of a complicated hull shape and the computed intersection between the hull and the free surface given in the thick curve. This example is a right skeg part of a twin-skeg vessel. This figure demonstrates that it is necessary to have a surface hull model for the efficient and accurate computation of the wetted part for seakeeping analysis.

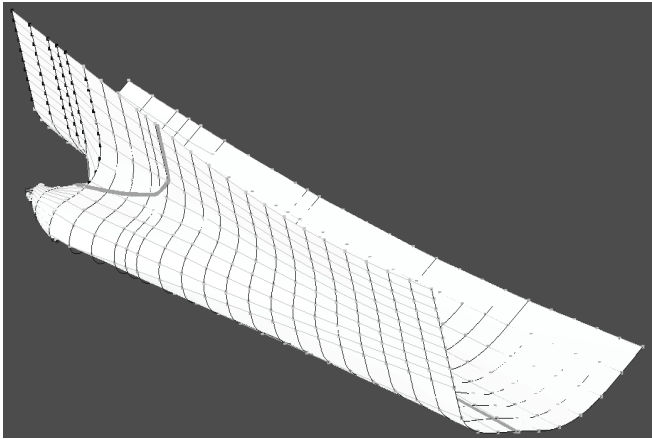


Fig. 10 The computed intersection curve between the hull and free water (The intersection between the hull and free surface as magenta curves).

The weakly nonlinear method requires either the panels for the entire hull shape or the wetted part at each computation step under various sea conditions depending on the detailed approach taken in the actual implementation. The latter case is also applied to the weak-scatterer analysis method. In this case, the motion of the ship changes as well as the waves near the ship and the wetted part of the hull changes accordingly at each computation step. For such analyses, the wetted part should be computed accurately and efficiently, which is almost impossible only by using the lines. However, when the hull shape is given in surfaces, the wetted part computation can be easily done.

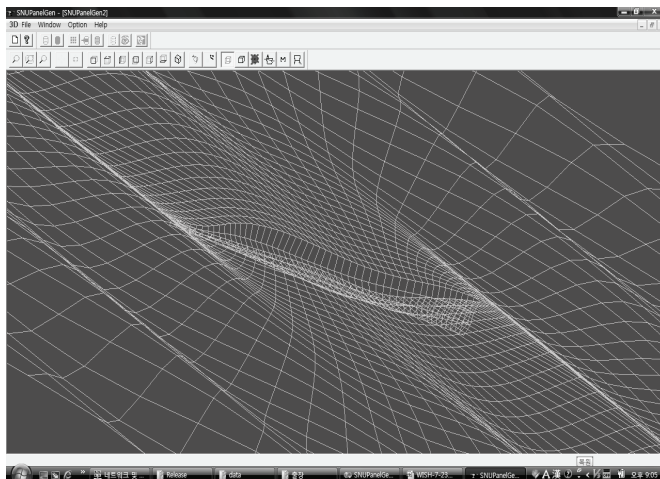


Fig. 11 An instance of generated panels with free surface (Ko et al., 2010).

The hull is translated and rotated to reflect the motion of the ship and the waves are given as a combination of mathematical functions. Then the intersection between the hull shape and the free surface representing the waves is computed. Fig. 11 shows an example of generated panels given a ship hull in waves. As demonstrated in the figure, the wetted part has been accurately obtained following the shape of the waves on the hull.

## CONCLUSIONS

Geometric modeling is an essential tool for handling a hull shape and provides useful operations for various purposes. Among them, surface modeling and surface-to-surface computation techniques are introduced and their application to ship hull design and analysis is presented. Since the hull shape is given as surfaces as oppose to curves in the lines drawings, various complicated computations can be performed accurately and efficiently. In particular seakeeping analysis is a primary beneficiary of this approach.

As more and more novel design criteria are considered, the hull shape becomes complicated. For example, twin-skeg, a moonpool or various appendages make the hull shape very complex to handle by using the lines. However, the surface model of such shapes, if modeled properly, can be efficiently processed for various computations. Therefore, the need for surface modeling of a ship hull increases.

Enhancing the surface modeling and SSI computation for a novel class of ships with complicated hull form is recommended for future work.

## ACKNOWLEDGEMENT

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2010-C1090-1031-0006)

## REFERENCES

- Bajaj, C.L. Hoffmann, C.M. Hopcroft, J.E. and Lynch, R.E., 1988. Tracing surface intersections. *Computer-Aided Geometric Design*, 5(4), pp. 285-307.
- Barnhill, R.E. and Kersey, S.N., 1990. A marching method for parametric surface/surface intersection. *Computer Aided Geometric Design*, 7(1-4), pp. 3-16.
- Hoschek, J. and Lasser, D., 1993. *Fundamentals of Computer Aided Geometric Design*. A. K. Peters, Wellesley, MA.

- Kim, K.H. and Kim, Y., 2009. Time-Domain Analysis of Nonlinear Ship Motion Responses Based on Weak-Scatterer Hypothesis. *19<sup>th</sup> International Offshore and Polar Engineering Conference*, International Society of Offshore and Polar Engineers, Osaka, Japan.
- Ko, K.H. Park, T. Kim, K.H. Kim, Y. and Yoon, D.H., 2010. *Development of Panel Generation System for Seakeeping Analysis*. Submitted for publication.
- Lane, J.M. and Riesenfeld, R.F., 1981. Bounds on a polynomial. *BIT: Nordisk Tidskrift for Informations-Behandling*, 21(1), pp. 112–117.
- Moore, R.E., 1966. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ.
- Patrikalakis, N.M. and Maekawa, T., 2001. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer-Verlag, New York, NY.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P., 1988. *Numerical Recipes in C*. 2<sup>nd</sup> Ed., Cambridge University Press, New York, NY.
- Rossignac, J.R. and Requicha, A.A.G., 1987. Piecewise-circular curves for geometric modeling. *IBM Journal of Research and Development*, 31(3), pp. 296–313.
- Tosaka, 2001. [http://commons.wikimedia.org/wiki/File:Ship\\_lines\\_\(3D\\_design\).PNG](http://commons.wikimedia.org/wiki/File:Ship_lines_(3D_design).PNG)
- Von Karman, T., 1929. *The impact of seaplane floats during landing*. NACA, Technical Note 321, Washington.
- Wagner, H., 1932. Über Stoss und Gleitvorgänge an der Oberfläche von Flüssigkeiten. *Zeitschr. F. Angewandte Mathematik und Mechanik*, 12(4), pp. 193–235.

## APPENDIX

Consider two surface patches  $\mathbf{r}_1(u, v)$  ( $0 \leq u, v \leq 1$ ) and  $\mathbf{r}_2(\sigma, t)$  ( $0 \leq \sigma, t \leq 1$ ) which intersect with each other. Then the three methods computing intersection are introduced. For more details, see (Patrikalakis and Maekawa, 2001).

### Lattice Method

The essence of the lattice method is to transform the surface-to-surface intersection problem into a set of intersection problems between a surface and a curve. Here, the curve is an iso-parametric curve obtained from an intersecting surface. Assume that  $\mathbf{c}_j(t) = \mathbf{r}_2(\sigma_j, t)$ ,  $\sigma_j$  is a constant. Then the intersection problem becomes finding the roots of the following equation.

$$\mathbf{r}_1(u, v) = \mathbf{c}_j(t) \text{ for } \sigma = \sigma_j. \quad (4)$$

This equation has three equations and three unknowns. So it can be easily solved by using Newton's method. By changing  $\sigma$  value, a set of intersection points on  $\mathbf{r}_1(u, v)$  is obtained, which are then connected to form an intersection curve.

### Subdivision Method

The pseudo-code for the subdivision method is as follows.

```

1: Intersection( $\mathbf{r}_1, \mathbf{r}_2, u_l, u_h, v_l, v_h$ )
2:   If( $\mathbf{r}_2(\sigma, t) \cap \mathbf{r}_1(u, v) == \emptyset$ )
3:     return
4:    $u_{half} = (u_l + u_h)/2$ ,  $v_{half} = (v_l + v_h)/2$ ,
5:   If( $|u_l - u_h| < TOL$  or  $|v_l - v_h| < TOL$ )
6:     Add  $\mathbf{r}_1(u_{half}, v_{half})$  to  $L$ 
7:     return
8:   Intersection( $\mathbf{r}_1, \mathbf{r}_2, u_l, u_{half}, v_l, v_{half}$ )
9:   Intersection( $\mathbf{r}_1, \mathbf{r}_2, u_{half}, u_h, v_l, v_{half}$ )
10:  Intersection( $\mathbf{r}_1, \mathbf{r}_2, u_l, u_{half}, v_{half}, v_h$ )
11:  Intersection( $\mathbf{r}_1, \mathbf{r}_2, u_{half}, u_h, v_{half}, v_h$ )
12:  return
13: end

```

Here,  $L$  the list containing the intersection points.  $TOL$  is the user-defined tolerance. The input of this code is  $\mathbf{r}_1, \mathbf{r}_2, u_l, u_h, v_l$  and  $v_h$ . Line 2 checks if the given domain defined by  $\mathbf{r}_1(u, v)$  ( $u_l \leq u \leq u_h, v_l \leq v \leq v_h$ ) intersects  $\mathbf{r}_2(\sigma, t)$ . If no intersection exists, the program returns. Otherwise, the process moves to Line 5. If the size is smaller than the tolerance, then the center of the domain is added to the list  $L$ . Otherwise, each subdivided domain is tested for intersection in a recursive manner. Once the program is terminated, the intersection points in the list  $L$  are obtained. Line 2 can be implemented by using interval arithmetic. Namely,  $E(u, v, \sigma, t) = |\mathbf{r}_1(u, v) - \mathbf{r}_2(\sigma, t)|$  is evaluated in interval arithmetic. If the resulting interval contains zero, then both surfaces intersect.

### Tracing Method

The intersection of  $\mathbf{r}_1(u, v)$  and  $\mathbf{r}_2(\sigma, t)$  is the root of the equation  $\mathbf{r}_1(u, v) = \mathbf{r}_2(\sigma, t)$ , which contains three equations and four unknowns. So it cannot be solved by an algebraic method. However, by introducing the arc length parametrization, a system of ODEs that are satisfied on the intersection is formulated as follows:

$$\sigma = \frac{d\sigma}{ds} = \frac{\text{Det}(\mathbf{h}, \mathbf{r}_{2t}, \mathbf{N}^{r_2})}{\mathbf{N}^{r_2} \cdot \mathbf{N}^{r_2}}, \quad (5)$$

$$t = \frac{dt}{ds} = \frac{\text{Det}(\mathbf{r}_{2\sigma}, \mathbf{h}, \mathbf{N}^{r_2})}{\mathbf{N}^{r_2} \cdot \mathbf{N}^{r_2}}, \quad (6)$$

$$u = \frac{du}{ds} = \frac{\text{Det}(\mathbf{h}, \mathbf{r}_{1v}, \mathbf{N}^{r_1})}{\mathbf{N}^{r_1} \cdot \mathbf{N}^{r_1}}, \quad \text{and} \quad (7)$$

$$v = \frac{dv}{ds} = \frac{\text{Det}(\mathbf{r}_{1u}, \mathbf{h}, \mathbf{N}^{r_1})}{\mathbf{N}^{r_1} \cdot \mathbf{N}^{r_1}}. \quad (8)$$

where  $s$  is the arc-length parameter, the subscripts indicate

partial derivatives,  $Det$  denotes the determinant, and

$$\mathbf{N}^s = \mathbf{r}_{2\sigma} \times \mathbf{r}_{2t}, \quad \mathbf{N}^i = \mathbf{r}_{1u} \times \mathbf{r}_{1v}. \quad (9)$$

The normal vectors for the surfaces are obtained at a point on the intersection. The vector  $\mathbf{h}$  is the tangent direction of the intersection curve at each point that is used as the marching direction of tracing. It is computed from the intersecting surfaces. However, depending on the intersection type, it needs to be computed differently. For a transversal intersection case where the normal vectors are not parallel, it can be computed by using Equation (10).

$$\mathbf{h} = \pm \frac{\mathbf{N}^{s1} \times \mathbf{N}^{s2}}{|\mathbf{N}^{s1} \times \mathbf{N}^{s2}|} \quad (10)$$

However, when two surfaces are intersecting tangentially, the two normal vectors are parallel to each other, namely, the tangent planes of both surfaces at the intersection coincide. In this case, a unique tangent vector  $\mathbf{h}$  cannot be determined using Equation (10).

So, higher order derivative terms should be used to determine  $\mathbf{h}$ . By using the fact that at the tangential intersection the normal curvatures are the same, a quadratic equation is derived as follows:

$$b_{11}(\sigma')^2 + 2b_{12}(\sigma')(t') + b_{22}(t')^2 = 0, \quad (11)$$

where  $b_{11}$ ,  $b_{12}$  and  $b_{22}$  are functions of the first and the second fundamental form coefficients of the given surfaces.

If  $b_{11} \neq 0$ ,

$$\mathbf{h} = \frac{v\mathbf{r}_{2\sigma} + \mathbf{r}_{2t}}{|v\mathbf{r}_{2\sigma} + \mathbf{r}_{2t}|}, \quad v = \frac{\sigma'}{t'}, \quad (12)$$

and if  $b_{11} = 0$  and  $b_{22} \neq 0$ ,

$$\mathbf{h} = \frac{\mathbf{r}_{2\sigma} + \mu\mathbf{r}_{2t}}{|\mathbf{r}_{2\sigma} + \mu\mathbf{r}_{2t}|}, \quad \mu = \frac{t'}{\sigma'}, \quad (13)$$