Kapal: Jurnal Ilmu Pengetahuan dan Teknologi Kelautan
(Kapal: Journal of Marine Science and Technology)
journal homepage: http://ejournal.undip.ac.id/index.php/kapal
2301-9069 (e)
1829-8370 (p)

Ship Hull Geometry Design By Using B-Spline Curve Fitting and Lofted Surface Generation

Sheely Leony Artha Pasaribu[1)], Berlian Arswendo Adietya[2)], Muhammad Iqbal[2)], Hasanudin[1)], Wiwin Sulistyawati[3)], Mohammad Sholikhan Arif[1,4)], Ardi Nugroho Yulianto[1)*)]

[1)] Department of Naval Architecture, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[2)] Department of Naval Architecture, Universitas Dipenegoro, Semarang, Indonesia
[3)] Department of Naval Engineering, Universitas Pembangunan Nasional Veteran Jakarta, Jakarta, Indonesia
[4)] Department of Naval Architecture, Ocean, and Marine Engineering, University of Strathclyde, Glasgow, United Kingdom
[*)] Corresponding Author: ardi.nugroho@its.ac.id

Article Info

Keywords:
Computer Aided Design (CAD);
Design;
Hydrostatic;
Ship Hull

Article history:
Received: 13/05/2026
Last revised: 29/06/2026
Accepted: 30/06/2026
Available online: 30/06/2026
Published: 01/07/2026

DOI:
https://doi.org/10.14710/kapal.v23i2.84399

Abstract

As one of the largest archipelago countries in the world, Indonesia requires technological capability to independently develop efficient ship design systems. Currently, the Computer-Aided Design (CAD) software used for domestic ship design is still largely dominated by foreign commercial products, which require costly subscription fees. This dependence limits the technological independence of the national shipbuilding industry. This study aims to develop a simple CAD-based program for representing ship hull geometry using a parametric mathematical approach. The program constructs sectional hull curves from input fit points obtained from hull offset data. Each section is represented using a B-spline parametric curve, where the curve parameterization, basis functions, and corresponding control points are calculated to reconstruct the sectional geometry within the developed CAD environment. The generated sectional curves are then discretized using an equal arc-length approach to obtain corresponding points between sections. These points are connected through a lofting procedure to form a continuous three-dimensional hull surface represented by a triangular mesh. The research methodology includes system design, curve and surface generation, geometric visualization, and validation through comparison with results obtained from commercial CAD software. The results demonstrate that the developed system is capable of accurately reconstructing sectional curves and generating a consistent lofted hull surface suitable for representing simple ship hull forms. This work provides an initial step toward the development of domestic ship hull design tools to support preliminary geometric modeling in Indonesia′ s maritime industry.

1. Introduction

Indonesia, as the largest archipelagic country in the world, has a very high dependence on the maritime sector, particularly in maritime transportation, the shipbuilding industry, and marine resource exploration [1]. The existence of thousands of islands distributed across the Indonesian archipelago makes ships the primary mode of transportation for both logistics distribution and community mobility.

Along with the rapid development of computer hardware and software technologies, as well as advances in Computer-Aided Design (CAD) systems, the representation of ship hull geometry has undergone significant transformation in the ship design process, including its role in improving stability evaluation and optimizing cargo space, which are essential aspects in ship design [2, 3]. Initially, hull representation relied primarily on curve-based approaches, but it has gradually evolved toward three-dimensional surface-based representations. Curve-based representations are still commonly used due to their relatively simple implementation; however, this approach has limitations in providing intuitive three-dimensional visualization and comprehensive surface information [4]. This condition has increased the demand for more efficient and digitally integrated design processes. Consequently, Computer-Aided Design (CAD) technology plays a crucial role as the backbone of ship hull geometry modeling [5].

In modeling ship hull forms, various mathematical methods have been developed to generate curves and surfaces, including B-Spline, Non-Uniform Rational B-Spline (NURBS), T-Spline, Bézier curves, cubic splines, and subdivision surfaces. Each of these approaches offers specific advantages in terms of shape flexibility, local control, and the capability to represent complex curved geometries. To date, several studies in Indonesia have attempted to develop ship design modeling programs.

However, these developments have not progressed significantly to the level of commercial viability for domestic use, and the resulting systems remain largely experimental or limited to academic applications.

Computer-Aided Design (CAD) is a computer-based system used to assist the processes of engineering design, modeling, and technical documentation in a digital environment. The early development of CAD can be traced back to the 1960s through the interactive computer graphics system "Sketchpad," developed by Ivan Sutherland [6]. Since then, the advancement of Computer-Aided Design and Computer-Aided Manufacturing (CAD/CAM) technologies has become a fundamental element in modern industries, including the shipbuilding sector. CAD/CAM software enables design, analysis, and manufacturing processes to be conducted with higher precision and efficiency. However, the situation in Indonesia indicates that technological independence in this field remains limited. The national industry continues to rely heavily on foreign commercial software such as AutoCAD, Siemens NX, Mastercam, Maxsurf, and NAPA. Although these applications provide advanced features, they require high licensing costs and are not always affordable for smaller shipyards or educational institutions.

In the development of geometry within Computer-Aided Design (CAD) systems, the ability to model curves with high precision and flexibility is a fundamental requirement, particularly in engineering industries such as automotive, architecture, and shipbuilding. To address this need, various curve-generation methods have been developed. Among the many available approaches, B-spline and Non-Uniform Rational B-Spline (NURBS) are currently the most widely used due to their capability to represent smooth curves with local control and their flexibility in forming complex and highly precise contours [7]. Therefore, spline-based geometry representation has become an important foundation in the development of CAD-based modeling systems.

Several previous studies have demonstrated the application of spline-based methods in ship geometric modeling. A bulbous bow modeling method using cubic B-spline curves was developed to construct a wire model, which was subsequently fitted into a B-spline surface [8]. A constrained hull design approach was further proposed using B-spline curves and surfaces, where station curves were generated based on design constraints and lofted to form the final hull surface [9] In addition, a mesh-based morphing method was introduced to rapidly generate hull form variations through interpolation between original and target hull surfaces [10]. These studies show that spline-based and computational geometry methods have been widely used to support efficient hull form generation and modification.

The development of Computer-Aided Design systems for geometric modeling using Bézier curves has long been applied in free-form surface design. In ship design applications, cubic Bézier curves combined with a curve-plane intersection method have been used to generate parametric bulbous bow geometry in a CAD-based solid modeling environment for CFD optimization [11]. A similar method was also applied to generate a parametric submarine hull form by considering geometric input parameters such as nose radius, tail radius, and length-to-height ratio [12]. These approaches indicate that parametric curve implementation is essential not only for surface modeling, but also for supporting automated solid model generation in ship design.

However, several studies emphasize the advantages of rational formulations by introducing weights at control points, which enhance the flexibility of geometric representation and enable the creation of more precise shapes [13]. Another common challenge in engineering geometry modeling is the difficulty of directly using trimmed NURBS models in Isogeometric Analysis (IGA). To address this issue, a conversion approach from NURBS to T-spline through reparameterization has been proposed. This method reconstructs interconnected patches with smooth continuity, thereby maintaining geometric consistency. Such an approach is capable of eliminating gaps between surfaces while preserving geometric accuracy [14].

The development of geometric modeling has also expanded beyond conventional curve and surface representations. Multi-sided surface patches have been introduced to handle non-standard surface domains in CAD environments [15]. Furthermore, volumetric modeling based on trimmed B-spline trivariates has been proposed to represent both the boundary and interior of three-dimensional objects [16]. More recently, generalized Bézier volumes have been developed for volumetric modeling over simple convex polyhedra, showing the continued development of Bézier-based representation in advanced geometric modeling [17]. In addition to geometric modeling, spline and NURBS formulations have also been applied in numerical analysis. A high-order panel method using quadratic geometry and quadratic basis functions was developed to improve the accuracy of curved geometry representation in potential flow analysis [18]. A B-spline-based higher-order panel method was also applied to marine propeller analysis by representing both geometry and solution using B-spline functions [19]. Furthermore, a desingularized high-order panel method based on NURBS was developed for three-dimensional potential flow problems, demonstrating that NURBS can provide accurate body geometry representation with a relatively small number of panels [20]. These studies indicate that parametric geometry is not only important for CAD modeling, but also for improving accuracy and efficiency in engineering analysis.

Due to the presence of weighted control points and non-uniform basis functions, NURBS representations exhibit relatively complex characteristics. Studies combining recursive evaluation techniques with Gauss–Legendre integral methods have demonstrated that this approach not only accelerates computational processes but also improves the accuracy of the resulting geometric representation. Furthermore, it supports the automation of hull line design with greater precision and efficiency [21]. Subsequent research has also introduced a flattening algorithm based on the bisection method. This algorithm significantly improves flattening accuracy without drastically increasing computational cost and demonstrates greater stability across various NURBS curve configurations compared with conventional flattening algorithms [22]. In addition, reparameterization of B-spline surfaces has been proposed as a solution to enhance control and efficiency in ship hull form design. Parameter transformation using monotonic functions shows that reparameterization produces hull geometries with improved surface quality, more uniform grid distribution, and the capability to perform localized design modifications without significantly affecting the overall hull form [4].

The studies discussed above represent several approaches that have been developed for ship hull form generation throughout the evolution of Computer-Aided Design (CAD) technology. Following the construction of curves in CAD, the next stage involves the development of surfaces capable of representing ship designs in three-dimensional form. Surface generation in CAD is therefore considered a continuation of curve construction. Differential geometry provides the mathematical framework for understanding the properties of curvature and surface continuity, which are subsequently implemented in surface-generation algorithms within CAD software. In practice, surface generation does not rely solely on control curves but must also satisfy requirements related to geometric continuity, curvature smoothness, and integration with other geometric elements.

By implementing existing curve-generation methods and incorporating automated calculations of ship hydrostatic properties, the program developed in this research is expected to provide an alternative to currently available commercial software. Although the present study focuses on the modeling of relatively simple ship hull forms, the research is directed toward developing a computational framework capable of representing ship hull geometry using numerical and parametric methods. Therefore, this research has three primary objectives: (1) to develop a Computer-Aided Design (CAD)-based program capable of generating and representing ship hull sectional geometry using a parametric curve formulation; (2) to implement a surface generation method through a lofting approach that constructs a continuous three-dimensional hull surface from sectional curves using equal arc-length discretization and triangular mesh representation; and (3) to verify and evaluate the accuracy of the implemented geometric modeling algorithms through comparison with results obtained from established commercial CAD software. Through these objectives, the study aims to demonstrate that the developed system is capable to reconstruct sectional curves and generating a consistent lofted hull surface suitable for representing ship hull geometry in preliminary design applications.

2. Method

2.1. Curve Generation

One of the curve-generation methods that has been developed for a considerable period is the Bézier curve, shown in Figure 1. Bézier curves represent a geometric modeling technique widely used in Computer-Aided Design (CAD) systems to construct free-form curves and surfaces. Among the various forms of Bézier curves, the cubic Bézier curve is one of the most commonly applied because it provides a balance between geometric flexibility and computational efficiency [13].

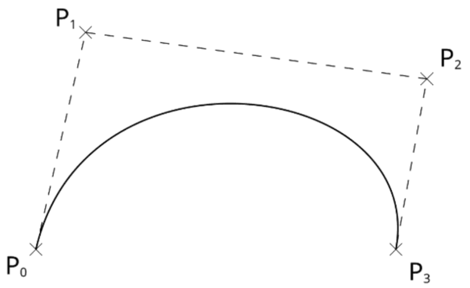


Figure 1 Bezier Curve

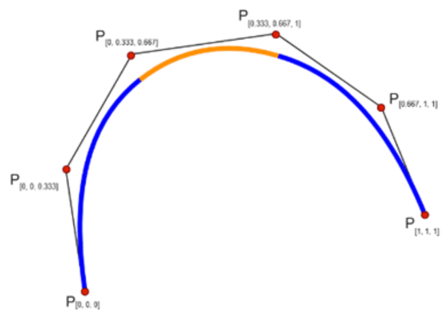


Figure 2 Cubic Spline

Another commonly used approach is the cubic spline method, shown in Figure 2. Cubic splines represent a numerical interpolation technique used to generate smooth curves that pass through a series of data points. Each segment of a cubic spline is constructed in such a way that the resulting curve maintains continuity up to the second derivative, thereby producing smooth transitions between adjacent data [23].

In its implementation, the cubic spline method constructs a system of simultaneous equations based on the continuity conditions of the function and its derivatives at the nodal points. For n data points, the method generates n − 1 spline functions, each defined over a specific interval. Each spline must satisfy four primary conditions: the function values must match the given data points, the first and second derivatives must be continuous at each connection point, and additional boundary conditions must be specified. These boundary conditions may take the form of a natural spline, in which the second derivatives at the endpoints are set to zero, or a clamped spline, in which the first derivatives at the endpoints are explicitly defined.

In contemporary CAD software development, B-Spline (Basis Spline) curves are among the most widely used parametric representations due to their flexibility and local control properties in geometric modeling. The term "B-Spline" refers to a combination of basis functions that collectively define a curve or surface constructed from a set of control points. This approach was developed to overcome the limitations of linear interpolation and high-degree polynomial curves, which tend to become numerically unstable as the number of control points increases [24]. In general, a B-Spline curve of degree k (order k + 1) can be expressed as follows:

$$C(u) = \sum_{i=0}^{n} N_{i,k}(u) \cdot P_i \tag{1}$$

Here, $C(u)$represents a point on the curve corresponding to the parameter u, $P_i$denotes the i-th control point, $N_{i,k}(u)$is the B-Spline basis function of order k, $u$is the parameter within the domain $u \in [u_0, u_m]$, and $n$represents the number of control points minus one.

The basis function $N_{i,k}(u)$is defined recursively using the Cox–de Boor algorithm as follows:

- For k=0

$$N_{i,0}(u) = \begin{cases} 1, & For\ u_i \leq u < u_{i+1} \\ 0, & Others \end{cases} \tag{2}$$

- For k>0

$$N_{i,0}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u) \tag{3}$$

To construct a B-Spline curve, the following components are required [24]:

a) Control points $P_0, P_1, \ldots, P_n$
b) Curve degree k
c) Knot vector $U = \{u_0, u_1, \ldots, u_{n+k+1}\}$

Figure 3 illustrates the visualization of a B-Spline curve generated from a sequence of control points. The B-Spline curve represents a flexible mathematical formulation for curved shapes and is widely applied in Computer-Aided Design (CAD), including in ship hull form modeling. B-Spline curves constitute an important foundation in the development of isogeometric methods, which enable direct integration between CAD-based geometric representation and numerical analysis. This integration strengthens their applicability in engineering design, including structural analysis of plate structures [25].

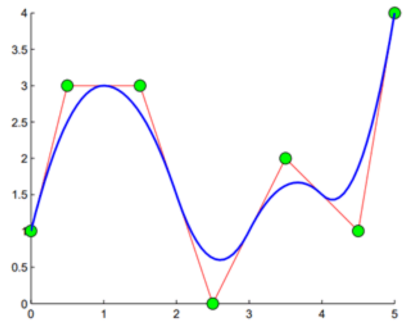


Figure 3 B-Spline Curve with Some Control Point

Furthermore, Non-Uniform Rational B-Splines (NURBS) represent a generalized form of parametric curves and surfaces that provide a high degree of flexibility and are widely used in Computer-Aided Design (CAD), including in the automotive, architectural, and shipbuilding industries. NURBS are an extension of B-Spline curves in which each control point is assigned a weight. The introduction of these weights enables the exact representation of conic sections such as circles, ellipses, and hyperbolas, which cannot be represented precisely using standard B-Spline formulations [24]. A NURBS curve of degree p, with $n + 1$control points and corresponding weights w_i, is defined as:

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) \cdot w_i \cdot P_i}{\sum_{i=0}^{n} N_{i,p}(u) \cdot w_i} \tag{4}$$

where $C(u)$denotes a point on the parametric curve, $P_i$represents the i-th control point, $w_i$is the weight associated with the i-th control point, $N_{i,p}(u)$denotes the B-Spline basis function of degree p, and $u$is the parameter defined within the domain $[u_0, u_m]$. NURBS can also be used to construct surfaces using two parametric variables, $u$and v, which are defined as follows:

$$S(u,v) = \frac{\sum_{i=0}^{n} \sum_{i=0}^{m} N_{i,p}(u) \cdot M_{j,q}(v) \cdot w_{ij} \cdot P_{ij}}{\sum_{i=0}^{n} \sum_{i=0}^{m} N_{i,p}(u) \cdot M_{j,q}(v) \cdot w_{ij}} \tag{5}$$

NURBS provide local control over geometric shapes through the use of weights and knot vector structures, while also supporting non-uniform polynomial degrees. This capability is particularly important in ship hull design, as it allows complex contours to be represented with high precision [7]. Figure 4 (a) illustrates a NURBS curve generated from several control points connected by a control polygon (shown in red). Unlike standard B-Spline curves, NURBS consider not only the positions of control points but also their associated weights, enabling highly flexible and precise curve representations, including conic sections such as circles and ellipses that cannot be represented using conventional B-Splines alone. Meanwhile, Figure 4 (b) presents a NURBS surface, which is an extension of the NURBS basis functions into two parametric directions. Such surfaces can accurately represent curved geometries, such as ship hull surfaces, due to their properties of continuity, smoothness, and local control.

In the context of engineering design and analysis, the use of NURBS enables a seamless transition from geometric design to numerical simulation through the Isogeometric Analysis (IGA) approach. This is possible because the same basis functions can be employed both in CAD-based geometric representation and in numerical computation [26].

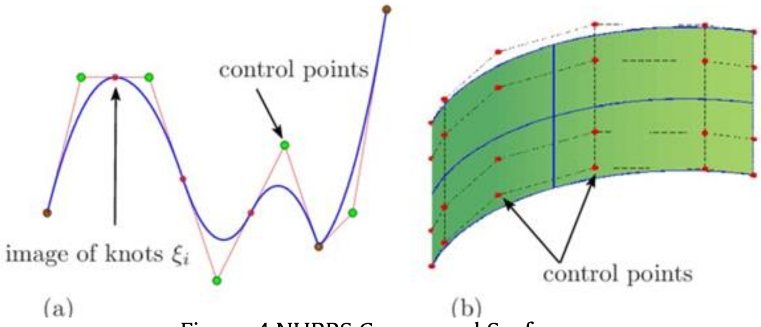


Figure 4 NURBS Curves and Surfaces

2.2. Solver Engine Program

For a given station s(for example, ST5), the user inputs a set of discrete points on the Y–Zsectional plane representing the hull cross-sectional shape at that station. The section position is assumed to remain constant at the longitudinal coordinate $x = x_s$, such that the sectional geometry is fully described in the transverse–vertical zycoordinates. The input data are expressed as:

$$Q_j^{(s)} = (z_j^{(s)}, y_j^{(s)}), j = 0,1 \dots, m \tag{6}$$

$$\tilde{Q}_j^{(s)} = (x_s, y_j^{(s)}, z_j^{(s)})$$

However, the solver operates entirely in the two-dimensional zy-space; the coordinate $x_s$is used only during the visualization or 3D reconstruction stage. The continuous section at station $s$is modeled as a B-spline curve of degree p as equation (4)

where $N_{i,p}(u)$denotes the B-spline basis function defined using the Cox–de Boor recursion, $\mathbf{P}_i^{(s)} = \left(Z_i^{(s)}, Y_i^{(s)}\right)$denotes the unknown control points, and $n + 1$is the number of control points. The knot vector $\mathbf{U} = \{U_0, \dots, U_{n+p+1}\}$ is selected as uniform (unclamped) over the parameter domain, which defines the range of $u_{\min}$and $u_{\max}$as follows $u_{\min} = U_p, u_{\max} = U_{n+1}$. Because the data points $\mathbf{Q}_j$are discrete, each data point is mapped to a parameter value u_j. The initial parameterization is performed using the chord-length or centripetal method with exponent α(commonly $\alpha = \frac{1}{2}$):

$$d_j = \| \mathbf{Q}_j - \mathbf{Q}_{j-1} \|^{\alpha}, j = 1, \dots, m \tag{7}$$

$$D = \sum_{j=1}^{m} d_j$$

$$t_0 = 0, t_j = \frac{\sum_{k=1}^{j} d_k}{D}, t_m = 1$$

These values are then mapped to the knot domain as $u_j = u_{\min} + t_j\,(u_{\max} - u_{\min})$. The purpose of this parameterization is to provide an initial estimate that is consistent with the geometric distribution of the data points, thereby ensuring that the least-squares system remains stable during the initial iteration. The control points $\{P_i\}$are determined by minimizing the squared error relative to the data points while incorporating a regularization term to reduce oscillations and prevent overfitting. The objective function is formulated as:

$$J(\mathbf{P}) = \sum_{j=0}^{m} \| \mathbf{C}(u_j) - \mathbf{Q}_j \|^2 + \mu \sum_{i=1}^{n-1} \| \Delta^2 \mathbf{P}_i \|^2 \tag{8}$$

where $\mu > 0$is a small regularization parameter, and the second-order finite difference operator is defined as $\Delta^2 P_i = P_{i-1} - 2P_i + P_{i+1}$. The first term ensures data fidelity, meaning that the fitted curve closely approximates the input data points. The second term acts as a roughness penalty, encouraging the resulting curve to remain smooth while preserving the primary geometric trend of the shape. To ensure that the curve passes exactly through the first and last data points, the following linear constraints are imposed $C(u_{\min}) = Q_0, C(u_{\max}) = Q_m$. Since $C(u)$is linear with respect to the control points P_i, the constraints can be expressed as:

$$\sum_{i=0}^{n} N_{i,p}\,(u_{\min})\, P_i = Q_0, \sum_{i=0}^{n} N_{i,p}\,(u_{\max})\, P_i = Q_m \tag{9}$$

These constraints ensure that the fitted B-spline curve interpolates the boundary data points while maintaining a smooth and stable geometric representation. The optimality system is solved using the Karush–Kuhn–Tucker (KKT) formulation:

$$\begin{bmatrix} \mathrm{A}^\top \mathrm{A} + \mu \mathrm{D}_2^\top \mathrm{D}_2 & \mathrm{C}^\top \\ \mathrm{C} & 0 \end{bmatrix} \begin{bmatrix} \mathrm{P} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathrm{A}^\top \mathrm{Q} \\ \mathrm{d} \end{bmatrix} \tag{10}$$

Because P consists of two components, $Z$and Y, the system can be solved either separately for each component or simultaneously in block form; mathematically, both approaches are equivalent. The initial parameter values $u_j$are generally not optimal; therefore, the fitting process can be improved through iterative reparameterization. For each internal data point $\mathrm{Q}_j (j = 1, \ldots, m-1)$, the closest parameter value on the curve is determined by solving:

$$u_j^{*} = \arg \min_{u \in [u_{\min}, u_{\max}]} \| \mathrm{C}(u) - \mathrm{Q}_j \|^2 \tag{11}$$

The iteration process is terminated when the relative error with respect to the geometric scale falls below a specified threshold. The Root Mean Square (RMS) error is defined as:

$$\mathrm{RMS} = \sqrt{\frac{1}{m+1} \sum_{j=0}^{m} \| \mathrm{C}(u_j) - \mathrm{Q}_j \|^2} \tag{12}$$

For the purposes of surface lofting and area calculation, the resulting curve can subsequently be resampled into points with an equal arc-length distribution (discussed separately in the surface construction method). Once the sectional curves at each station have been obtained through the B-spline fitting process, the next stage is to construct the three-dimensional hull surface between stations using the loft surface method. This surface is then used to compute the hull surface area, submerged volume, and other hydrostatic parameters. Suppose two consecutive stations along the longitudinal direction are $s_k$and s_{k+1}, located at positions $x_{s_k}, x_{s_{k+1}}$. The sectional curves at each station are expressed as parametric functions:

$$\mathrm{C}^{(s)}(u) = \begin{bmatrix} x_s \\ y^{(s)}(u) \\ z^{(s)}(u) \end{bmatrix} \tag{13}$$

where $u \in [u_{\min}, u_{\max}]$. To establish point correspondence between the two stations, each curve is first resampled using an equal arc-length distribution. The arc length of the sectional curve at station $s$is defined as:

$$L_s = \int_{u_{\min}}^{u_{\max}} \sqrt{\left(\frac{dy}{du}\right)^2 + \left(\frac{dz}{du}\right)^2}\, du \tag{14}$$

To generate consistent panels between stations, the arc length is divided into Nequal segments, where Nrepresents the desired number of panels (for example, $N = 1000$). The sampling points along the curve are therefore defined as $s_i = i\Delta s, i = 0, \ldots, N$, where $\Delta s = \frac{L}{N}$and Ldenotes the total arc length of the curve. The corresponding parameter values are obtained through the inverse arc-length function $u_i = \ell^{-1}(s_i)$. Thus, the points on the sectional curve can be expressed as:

$$\mathrm{P}_{s,i} = \begin{bmatrix} x_s \\ y^{(s)}(u_i) \\ z^{(s)}(u_i) \end{bmatrix} \tag{15}$$

The total number of sampled points is therefore $N + 1$. The surface between two adjacent stations is constructed by connecting points that share the same arc-length index. For two consecutive stations $s_k$and s_{k+1}, define the following points:

$$\begin{aligned} A_i &= \mathrm{P}_{s_k,i} \\ B_i &= \mathrm{P}_{s_k,i+1} \\ D_i &= \mathrm{P}_{s_{k+1},i} \\ C_i &= \mathrm{P}_{s_{k+1},i+1} \end{aligned}$$

These four points form a quadrilateral panel (quad panel) on the hull surface. Each quadrilateral panel is subsequently triangulated into two triangular elements:

$$\begin{aligned} T_{1,i} &= (A_i, B_i, C_i) \\ T_{2,i} &= (A_i, C_i, D_i) \end{aligned}$$

The area of a triangle in three-dimensional space is calculated using the norm of the cross product, illustrated in Figure 5 Accordingly, the half-hull surface area between two stations is obtained by summing the areas of all triangular panels:

$$A_{half} = \sum_{i=0}^{N-1} [A_{\triangle}(A_i, B_i, C_i) + A_{\triangle}(A_i, C_i, D_i)] \tag{16}$$

Following the construction of the hull surface, the hydrostatic calculations are subsequently performed. These calculations follow the formulations presented in the previous subsections, where numerical integration techniques are applied to the sectional data obtained from the geometric model. In particular, Simpson′ s 1/3 Rule is employed to integrate the sectional areas along the longitudinal axis of the ship in order to determine the displacement volume, Longitudinal Center of Buoyancy (LCB), and Waterplane Area (WPA). Through this approach, the hydrostatic parameters can be derived directly from the generated hull geometry, enabling efficient evaluation of the vessel′ s buoyancy characteristics and preliminary stability performance within the CAD-based modeling environment.

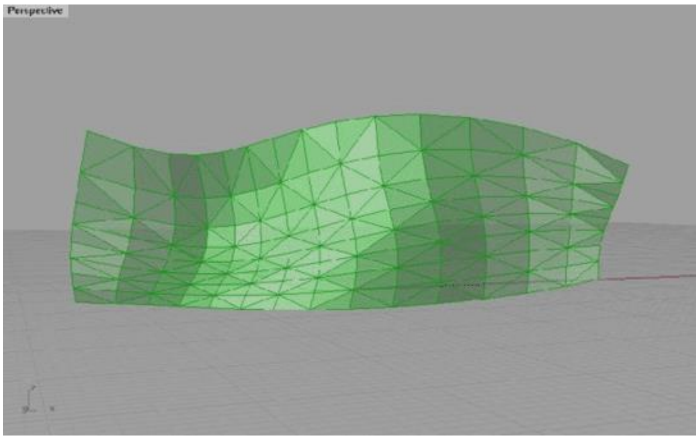


Figure 5 Triangular Panel

2.3. Program Development

The programming language employed in this research is Python. Python is a high-level programming language that has been widely recognized as an effective and efficient tool for handling repetitive computational tasks and automating engineering workflows [27]. Owing to its clear syntax, extensive standard library, and broad ecosystem of scientific and numerical packages, Python has become increasingly popular in the development of engineering applications, including software for ship geometry modeling and hydrostatic analysis. To provide an accessible and user-friendly interface, the program was developed with Tkinter as the Graphical User Interface (GUI) framework. Tkinter is the standard GUI library in Python and enables the creation of interactive desktop-based applications through components such as windows, buttons, input fields, menus, and dialog boxes. The use of Tkinter allows the developed program to present a visual working environment in which users can input hull parameters, define sectional points, execute curve-fitting and surface-generation procedures, and directly view the resulting outputs in an organized manner.

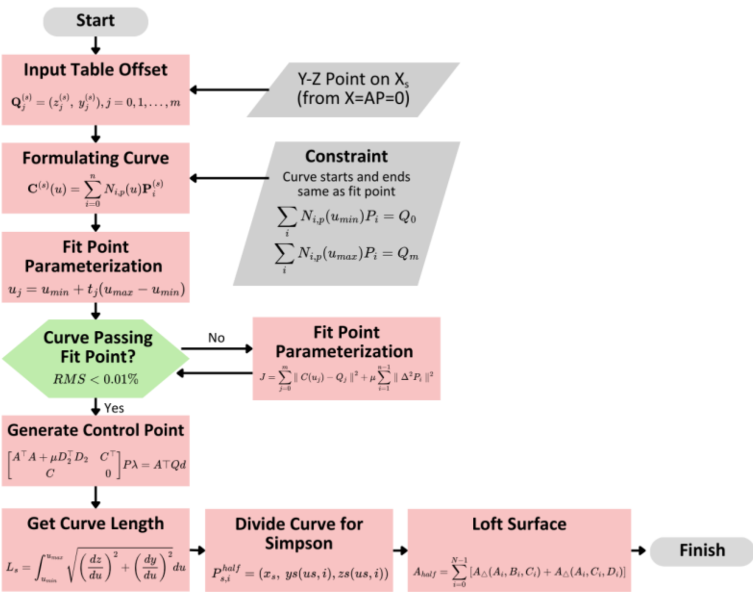


Figure 6 Program's Solver Engine

As shown in Figure 6, the computational workflow begins by defining the longitudinal positions of the stations relative to the Aft Perpendicular (AP), expressed in terms of the coordinate x. At each station, the user inputs pairs of sectional points on the $Y-Z$plane representing the geometric shape of the hull cross-section. These points are stored as fit points, which serve as the fundamental data for constructing the sectional curve. Based on this input data, a curve-fitting process is performed using parametric curves to obtain a continuous mathematical representation of the hull section. After the initial curve is generated, a parameter refinement stage is conducted to improve the conformity of the curve to the input fit points. This refinement can be achieved either through an iterative optimization process or through manual adjustment of the control points by the user when necessary.

The fitted curve is subsequently used to generate a more detailed geometric representation. At this stage, the arc length of the sectional curve is calculated and used to divide the curve evenly according to the equal arc-length distribution. This process produces a set of discrete points with high resolution, typically exceeding one thousand segments. These resampled points serve as the basis for establishing geometric correspondence between adjacent stations. By connecting points that share the same arc-length index on two neighboring stations, a lofted surface is constructed and represented as a triangular mesh. This discretized surface then becomes the geometric model of the ship hull, which can be used for further analysis.

3. Results and Discussion

3.1. Software Engine Development

The initial stage of this research focuses on adjusting the curve-generation algorithm so that it can represent ship sectional shapes stably and accurately from discrete point data. At this stage, a parametric B-spline curve-fitting method is developed using a least-squares approach combined with a reparameterization process. This approach allows the input data points (fit points) from table offset to differ from the control points, enabling the resulting curve to achieve a smoother representation while still preserving the overall geometric tendency of the sectional shape.

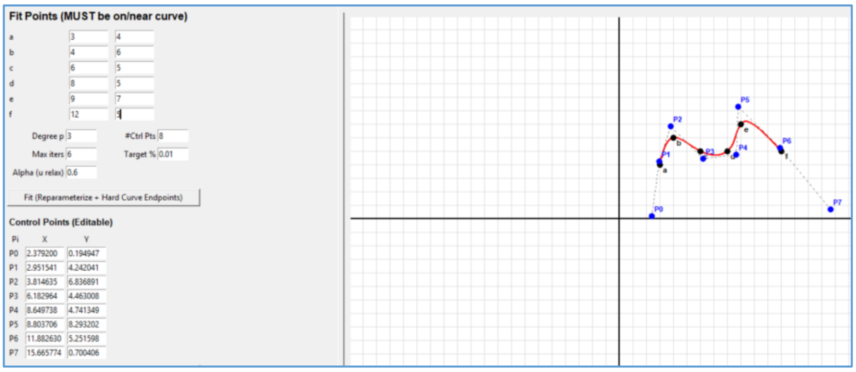


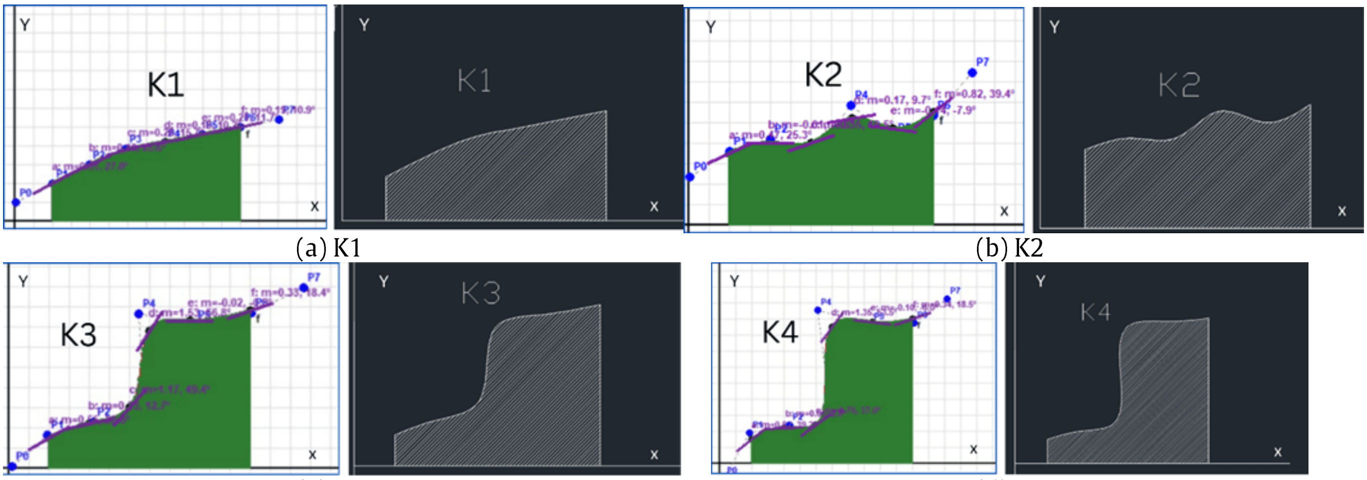
Figure 7. Difference of Fit Point (Black Dots) and Control Point (Blue Dots)

The distinction between fit points and control points is an important characteristic of B-spline curves. In this formulation, the shape of the curve is controlled through the positions of the control points rather than forcing the curve to pass exactly through every data point. Consequently, the curve can maintain geometric smoothness and numerical stability while still providing accurate approximation of the original sectional geometry as shown in Figure 7.

The initial implementation results of the proposed algorithm demonstrate that the generated curve is capable of following the distribution of the fit points in a stable manner, as illustrated in Figure 7. The figure shows that the resulting curve closely follows the pattern of the data points, while the control points serve as the structural framework that governs the shape of the curve. This indicates that the applied fitting method is capable of producing a smooth curve while simultaneously preserving the geometric characteristics of the sectional data.

3.2. Area Calculation Correction

The area correction with respect to the baseline $y = 0$ was conducted as an initial step to verify the accuracy of the implemented area calculation algorithm in the developed program. The area calculation was performed using the Simpson 1/3 numerical integration method, applied to the B-spline curve generated from the fitting of points defined within the application. The evaluation was carried out using five different curve shapes with varying slope characteristics, ranging from gently sloped curves to curves with very steep or nearly vertical gradients, as illustrated in Figure 8. This variation in slope was selected because the Simpson integration method used in the numerical computation is sensitive to changes in curve gradients, particularly in regions where the change of $y$with respect to $x$occurs rapidly.



(a) K1 (b) K2

(c) K3 (d) K4

Figure 8. Curve Area Above $y = 0$

Based on the comparison results presented in Table 1, it can be observed that for curves with gentle to moderate slopes (K1 and K2), the difference between the area calculated by the developed application and the area obtained from AutoCAD is very small. For curve K1, the difference is close to 0.00%, indicating excellent agreement between the two calculation methods. For curve K2, the difference reaches approximately 0.38%. For curves with locally steeper gradients (K3 and K4), the area difference begins to increase but still remains within an acceptable range for geometric analysis. The differences range between 0.52% and 0.70%, indicating that the numerical integration method is still capable of representing the curve area adequately even in the presence of steeper segments. In segments where the gradient is very steep, small changes in $x$correspond to large variations in y, which may increase local discretization errors compared to the adaptive integration techniques commonly implemented in commercial CAD software.

Table 1 Area above $y = 0$

Curve	Characteristic	Maximum Slope (degrees)	Developed Program Area (mm²)	Commercial Program Area (mm²)	Difference (%)
K1	Gentle	26.98	38.3056	38.3047	0.00%
K2	Moderate	39.38	44.6794	44.7018	0.05%
K3	Locally Steep	56.75	50.3094	50.5747	0.52%
K4	Very Steep	53.53	57.594	57.1964	-0.70%

Nevertheless, the overall results indicate that the accuracy of the developed program remains very good for curves with slopes below approximately 50° relative to the horizontal axis, where the area differences remain small and stable. This range of slope conditions is commonly encountered in most conventional ship hull cross-sections, particularly in the midship and bow regions where extreme gradient variations are generally absent.

These findings are also consistent with previous CAD-based ship geometry studies. Chrismianto and Kim (2014) demonstrated that curve parameterization using cubic Bézier curves and curve-plane intersection can be used to automatically generate parametric bulbous bow geometry. Similarly, Pérez and Clemente (2011) showed that B-spline fitting of station data followed by lofting can produce a complete hull surface representation. Compared with these studies, the present work emphasizes the importance of defining curve parameterization, control points, and corresponding discretized points properly, since these factors directly influence the accuracy of sectional area calculation and the consistency of the generated lofted surface.

Therefore, the implemented area calculation method can be considered sufficiently accurate for CAD-based ship hull geometry modeling, particularly during the early stages of hull form analysis. The relatively small deviation compared to commercial software results demonstrates that the proposed approach can be reliably used as a foundation for subsequent hydrostatic calculations, such as displacement volume, cross-sectional area, and ship stability parameters.

3.3. User Interface for User Accessibility

After the fundamental algorithm was considered sufficiently accurate, the development was extended to the user interface (UI) aspect to facilitate the verification and testing process of the model. The interface allows users to directly input sectional points, perform curve-fitting operations, manually modify the positions of control points, and re-evaluate the resulting curve geometry. This functionality significantly supports the process of geometric verification and result validation, particularly during the early stages of development when multiple variations of sectional shapes need to be tested efficiently. By enabling interactive modification and immediate visualization of the fitted curves, the interface enhances the usability of the developed system and allows users to assess the geometric behavior of the model in a more intuitive manner. The design of the graphical user interface developed for this program is illustrated in Figure 9.

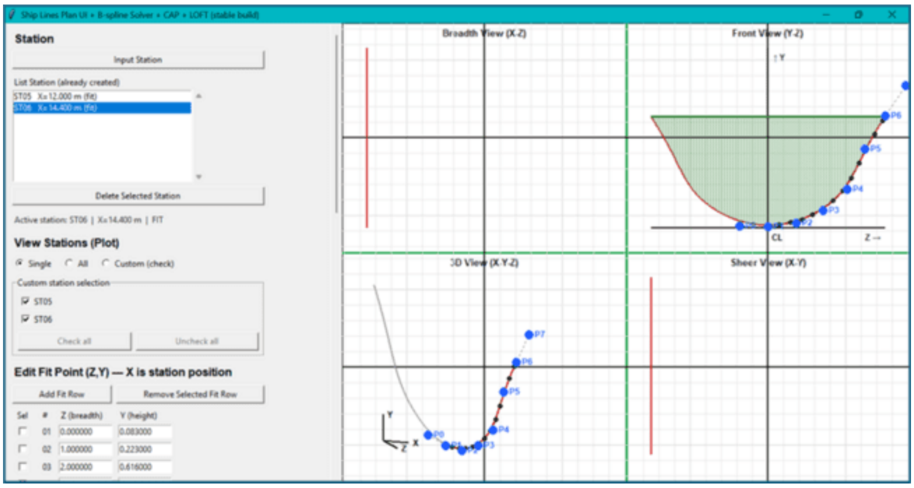


Figure 9. User Interface for User Accesbility

3.4. Cross-Section and Lofted Area

The next stage of development involves the implementation of a hull surface generation module. The sectional curves obtained from multiple stations are discretized into a number of points based on equal arc-length distribution, ensuring a consistent point distribution across each station. This discretization process produces a uniform set of sampling points along each sectional curve.

These corresponding points are then connected between adjacent stations to construct a lofted surface, which is represented as a triangular mesh. This discrete representation of the hull surface enables the three-dimensional geometry to be processed and analyzed numerically in a stable and computationally efficient manner. The resulting surface mesh serves as the geometric basis for subsequent hydrostatic calculations and visualization of the hull form, as illustrated in Figure 10.

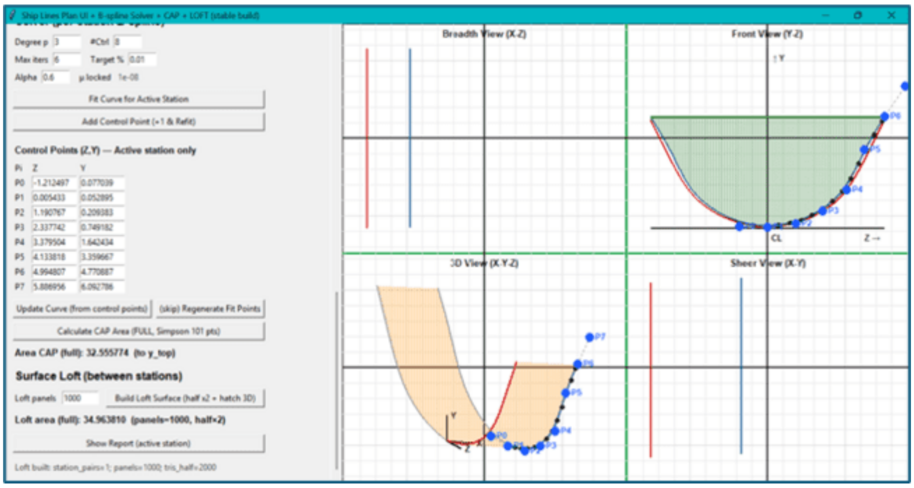
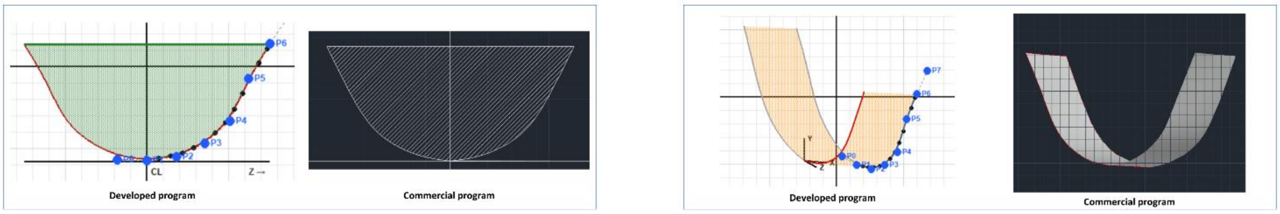


Figure 10. Lofted Surface

After the surface model has been successfully constructed, the system is subsequently used to compute the cross-sections between stations. The results from this stage indicate that the arc-length–based curve discretization method produces a more uniform distribution of panels compared to discretization based solely on the curve parameter. Consequently, this approach generates a lofted surface that is numerically more stable and geometrically consistent. This result is in line with previous studies in CAD-based three-dimensional geometry generation. Chrismianto and Kim (2014) used curve parameterization and curve-plane intersection to generate parametric bulbous bow geometry, while Pérez and Clemente (2011) generated a hull surface through B-spline station curves followed by a lofting process. Therefore, the present result confirms that proper curve parameterization and point correspondence between sections are important requirements for producing a consistent three-dimensional hull surface.

Before proceeding to the hydrostatic calculations, **Error! Reference source not found.** presents a comparison between the area calculations obtained from the sectional models shown in Figure 11 the corresponding results produced by widely used commercial CAD software. This comparison serves as an additional validation step to evaluate the accuracy and reliability of the developed computational method.



(a) Cross section at ST 6 (b) Area from ST 5- ST 6

Figure 11. Area Checking

4. Conclusion

This study developed a Computer-Aided Design (CAD)-based program for representing ship hull geometry using a parametric B-spline curve formulation combined with numerical surface generation techniques. The implemented method successfully reconstructed sectional curves from discrete data (table offset) and generated a continuous hull surface through a lofting approach based on equal arc-length discretization and triangular mesh representation. Validation through comparison with commercial CAD software indicates that the geometric results show good accuration, demonstrating the reliability of the implemented curve fitting and surface generation algorithms for preliminary hull modeling. Future work will focus on extending the current framework to represent the complete ship hull geometry and to integrate hydrostatic computation modules, including calculations of displacement, longitudinal center of buoyancy (LCB), and waterplane area (WPA). These developments are expected to enhance the capability of the system as a practical CAD-based tool for early-stage ship hull design and hydrostatic analysis.

Acknowledgements

The author greatly acknowledges the financial support that comes from the Ministry of Higher Education, Science and Technology of Republik Indonesia, scheme Penelitian Fundamental Reguler, under contract no. 2231/PKS/ITS/2026.

References

[1] Badan Pembinaan Hukum Nasional, "Indonesia Merupakan Negara Kepulauan yang Terbesar di Dunia," Oct. 28, 2015. [Online]. Available: https://bphn.go.id/berita-utama/indonesia-merupakan-negara-kepulauan-yang-terbesar-di-dunia-3441

[2] Hasanudin, A. D. Saputra, and A. N. Yulianto, "A Study on Stability and Passenger Capacity of Traditional River and Lake Boats," *BIO Web of Conferences*, vol. 157, 2025, doi: https://doi.org/10.1051/bioconf/202515703005

[3] A. N. Yulianto, R. R. Winarko, S. L. A. Pasaribu, R. F. Ahmad, B. A. Adietya, and Hasanudin, "Optimization of Inner Bilge Design to Enhance Cargo Tank Capacity in Tankers," *Kapal: Jurnal Ilmu Pengetahuan dan Teknologi Kelautan*, vol. 23, no. 1, pp. 49–63, 2026, doi: https://doi.org/10.14710/kapal.v23i1.82683

[4] L. Y. Wang and Z. J. Wang, "Reparameterization of B-spline surface and its application in ship hull form design," *Computer-Aided Design*, vol. 115, pp. 102–112, 2019, doi: https://doi.org/10.1016/j.oceaneng.2023.115535

[5] A. F. Molland, T. S. Turnock, and H. D. Hudson, *Ship Resistance and Propulsion: Practical Estimation of Propulsive Power*. New York, NY, USA: Cambridge University Press, 2011.

[6] I. E. Sutherland, "Sketchpad: A Man–Machine Graphical Communication System," Doctoral dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 1963.

[7] D. F. Rogers, *An Introduction to NURBS: With Historical Perspective*. Annapolis, MD, USA: Academic Press, 2001.

[8] F. Pérez, J. A. Suárez, J. A. Clemente, and A. Souto, "Geometric modelling of bulbous bows with the use of non-uniform rational B-spline surfaces," *Journal of Marine Science and Technology*, vol. 12, no. 2, pp. 83–94, 2007, https://doi.org/10.1007/s00773-006-0225-6

[9] F. Pérez and J. Clemente, "Constrained design of simple ship hulls with B-spline surface," *Computer-Aided Design*, vol. 43, no. 12, pp. 1829–1840, 2011, doi: https://doi.org/10.1016/j.cad.2011.07.008

[10] J. Y. Kang and B. Lee, "Mesh-based morphing method for rapid hull form generation," *Computer-Aided Design*, vol. 42, no. 11, pp. 970–976, 2010, doi: https://doi.org/10.1016/j.cad.2009.07.00

[11] D. Chrismianto and D.-J. Kim, "Parametric bulbous bow design using the cubic Bezier curve and curve-plane intersection method for the minimization of ship resistance in CFD," *Journal of Marine Science and Technology*, vol. 19, no. 4, pp. 479–492, 2014, doi: https://doi.org/10.1007/s00773-014-0278-x

[12] D. Chrismianto, A. F. Zakki, B. Arswendo, and D. J. Kim, "Development of Cubic Bezier Curve and Curve-Plane Intersection Method for Parametric Submarine Hull Form Design to Optimize Hull Resistance Using CFD," *Journal of Marine Science and Application*, vol. 14, no. 2, pp. 132–144, 2015, doi: https://doi.org/10.1007/s11804-015-1324-8

[13] J. K. Chau and S. W. Chau, "Computer-Aided Geometric Design and Panel Generation for Hull Forms Based on Rational Cubic Bézier Curves," *Computer Aided Geometric Design*, vol. 10, no. 5, pp. 431–447, 1993, doi: https://doi.org/10.1016/0167-8396(93)90030-7

[14] K. Kostas, "Ship-hull shape optimization with a T-spline based method," *Computer Methods in Applied Mechanics and Engineering*, vol. 279, pp. 415–438, 2014, doi: https://doi.org/10.1016/j.cma.2014.10.030

[15] P. Charrot and J. A. Gregory, "A pentagonal surface patch for computer aided geometric design," *Computer Aided Geometric Design*, vol. 1, no. 1, pp. 87–94, 1984, doi: https://doi.org/10.1016/0167-8396(84)90006-2

[16] F. Massarwi and G. Elber, "A B-spline based framework for volumetric object modeling," *Computer-Aided Design*, vol. 78, pp. 36–47, 2016, doi: https://doi.org/10.1016/j.cad.2016.05.003
[17] K. Qin, Y. Li, and C. Deng, "Generalized Bézier volumes over simple convex polyhedra," *Computer Aided Geometric Design*, vol. 111, p. 102338, 2024, doi: https://doi.org/10.1016/j.cagd.2024.102338
[18] D. J. Willis, J. Peraire, and J. K. White, "A Quadratic Basis Function, Quadratic Geometry, High Order Panel Method," in *Proc. 44th AIAA Aerospace Sciences Meeting*, 2006, pp. 1–17, doi: https://doi.org/10.2514/6.2006-1253
[19] G.-D. Kim, C.-S. Lee, and J. Kerwin, "A B-spline based higher order panel method for analysis of steady flow around marine propellers," *Ocean Engineering*, vol. 34, no. 15–16, pp. 2045–2060, 2007, doi: https://doi.org/10.1016/j.oceaneng.2007.02.013
[20] Z.-l. Gao and Z.-j. Zou, "A Three-Dimensional Desingularized High Order Panel Method Based on NURBS," *Journal of Hydrodynamics, Ser. B*, vol. 20, no. 2, pp. 137–146, 2008, doi: https://doi.org/10.1016/S1001-6058(08)60039-6
[21] X. Jiang and Y. Lin, "Relevant integrals of NURBS and its application in hull line element design," *Ocean Engineering*, vol. 251, p. 111147, May 2022, doi: https://doi.org/10.1016/j.oceaneng.2022.111147.
[22] K.-G. Zhu, G.-Y. Shi, and J. Liu, "Improved flattening algorithm for NURBS curve based on bisection feedback search algorithm and interval reformation method," *Ocean Engineering*, vol. 247, p. 110635, Mar. 2022, doi: https://doi.org/10.1016/j.oceaneng.2022.110635.
[23] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, 7th ed. New York, NY, USA: McGraw-Hill Education, 2015.
[24] L. Piegl and W. Tiller, *The NURBS Book*, 2nd ed. Berlin, Germany: Springer, 1997.
[25] T. H. Chien, *Development of Isogeometric Finite Element Methods*. Vietnam: Ton Duc Thang University, 2015.
[26] V. P. Nguyen, K. P. Nguyen, and B. S. A. Nguyen, "Two- and three-dimensional isogeometric cohesive elements for composite delamination analysis," *Composites Part B: Engineering*, vol. 53, pp. 390–408, 2013, doi: https://doi.org/10.1016/j.compositesb.2013.12.018
[27] M. Lutz, *Learning Python*, 5th ed. Sebastopol, CA, USA: O' Reilly Media, Inc., 2013.