

Tự động hóa rời rạc hình học vỏ tàu thủy

Automation of ship hull geometry discretization

>TS LÊ VĂN TOÀN*, THS ĐOÀN TRUNG VIỆT

Nhóm nghiên cứu AIT, Trường Đại học Giao thông vận tải TP. HCM

Email: *toan.le@ut.edu.vn

TÓM TẮT

Nghiên cứu này trình bày phương pháp rời rạc hóa hình học vỏ tàu thủy sử dụng lý thuyết toán học kết hợp với ngôn ngữ lập trình Python. Quá trình rời rạc này được triển khai nhằm bảo đảm rằng các đặc tính hình học của vỏ tàu không bị thay đổi khi chuyển ngược lại thành dạng 3D mặt (3D face). Kết quả của quá trình rời rạc hóa không chỉ tạo điều kiện thuận lợi cho việc thực hiện các điều chỉnh hình học trong các tính toán tối ưu mà còn tiết kiệm đáng kể thời gian xử lý. Đồng thời, các công đoạn xác nhận biên dạng tiết diện bất kỳ, phục vụ cho công tác hạ liệu và phóng dạng được thực hiện dễ dàng nhờ vào thông tin số hóa từ dữ liệu điểm. Phần mềm phát triển từ phương pháp này không chỉ giảm thiểu sai sót mà còn góp phần nâng cao mức độ tự động hóa và tối ưu hóa trong quy trình thiết kế và sản xuất vỏ tàu thủy.

Từ khóa: Rời rạc hóa hình học, vỏ tàu thủy, dữ liệu điểm, ngôn ngữ lập trình Python, tối ưu hóa hình học, phóng dạng, hạ liệu, tự động hóa thiết kế tàu.

ABSTRACT

This study introduces a novel method for discretizing ship hull geometry by integrating mathematical theory with the Python

programming language. The proposed approach ensures that the geometric properties of the hull remain intact when converted back into a 3D surface model. This precise discretization process facilitates efficient geometric adjustments during optimization calculations, significantly reducing processing time. Moreover, the digitized point cloud data simplifies the validation of cross-sectional profiles, enhancing the efficiency and reliability of material nesting and unfolding tasks. The developed software not only minimizes errors but also elevates the level of automation and optimization in ship hull design and manufacturing. By maintaining the fidelity of the original geometry, this method supports precise and adaptable hull modifications while streamlining key production processes. As the demand for intelligent and automated solutions in the shipbuilding industry continues to grow, this approach provides a robust foundation for improving efficiency, accuracy and innovation, ultimately contributing to the modernization and competitiveness of ship design and construction.

Keywords: Discretization of geometry, ship hull, point cloud, Python programming language, geometry optimization, unfolding, material nesting, ship design automation.

1. ĐẶT VẤN ĐỀ

Trong ngành công nghiệp đóng tàu, việc thiết kế và chế tạo vỏ tàu đóng vai trò quan trọng trong việc đảm bảo chất lượng và hiệu suất của tàu trong quá trình vận hành. Trong đó, các công đoạn phóng dạng và hạ liệu là những bước đầu tiên trong quy trình gia công, ảnh hưởng trực tiếp đến chất lượng sản phẩm cuối cùng và chi phí sản xuất. Tuy nhiên, các công đoạn này hiện nay chủ yếu được thực hiện thủ công, dẫn đến nhiều hạn chế về thời gian, chi phí và độ chính xác trong quá trình sản xuất.

Để giải quyết vấn đề này, nghiên cứu tập trung vào việc phát triển phương pháp rời rạc hóa hình học 3D vỏ tàu thủy thành dữ liệu điểm (point cloud) dựa trên lý thuyết toán học kết hợp ngôn ngữ lập trình Python. Phương pháp này không chỉ đảm bảo tính chính xác cao trong quá trình chuyển đổi từ hình học 3D sang dữ liệu điểm

mà còn giúp duy trì được các đặc tính hình học vỏ tàu khi chuyển ngược lại từ dữ liệu điểm sang hình học 3D. Nhờ vào kết quả rời rạc hóa, việc điều chỉnh hình học trong các tính toán tối ưu sẽ trở nên dễ dàng hơn, tiết kiệm thời gian và chi phí. Ngoài ra, các công đoạn xác nhận biên dạng tiết diện bất kỳ, phục vụ cho công tác phóng dạng và hạ liệu cũng trở nên thuận tiện hơn thông qua các thông tin số hóa đường hình nhận được.

Nghiên cứu này không chỉ góp phần nâng cao hiệu quả của các công đoạn trong quy trình đóng tàu mà còn mở ra khả năng ứng dụng rộng rãi trong các ngành công nghiệp khác, nơi yêu cầu các quy trình thiết kế và chế tạo chính xác, tiết kiệm và hiệu quả. Phương pháp rời rạc hóa hình học 3D này có thể giúp nâng cao sự tự động hóa trong các quy trình sản xuất, giảm thiểu sai sót, đồng thời tối ưu hóa thiết kế hình học vỏ tàu, góp phần cải thiện toàn bộ quy trình đóng tàu trong bối cảnh công nghiệp 4.0.

2. PHƯƠNG PHÁP LUẬN

*** Rời rạc hóa hình học 3D**

Trong nghiên cứu này, việc chuyển đổi dữ liệu hình học 3D của vỏ tàu thủy lưu dưới định dạng STL thành tập hợp điểm bề mặt (point cloud) với các tọa độ (x,y,z) được xác định bằng cách dò tìm giao điểm giữa tia dò và bề mặt mô hình, kết quả được lưu dưới dạng tệp CSV hoặc excel.

Đầu tiên, tiến hành xác lập tọa độ gốc (origin) và chuẩn hóa mô hình: Tọa độ gốc được đặt tại trung tâm của mặt đáy mô hình (mặt phẳng Oxy với z = 0) giúp đơn giản hóa tính toán và xử lý sau này và đồng nhất hóa hệ tọa độ mô hình 3D khi xuất dữ liệu.

Giả sử kích thước miền không gian bao quanh mô hình được ký hiệu:

$$[x_{\min}, y_{\min}, z_{\min}] \tag{1}$$

$$[x_{\max}, y_{\max}, z_{\max}] \tag{2}$$

Tọa độ gốc (origin) tính như sau:

$$(origin)_x = \frac{x_{\max} + x_{\min}}{2} \tag{3}$$

$$(origin)_y = \frac{y_{\max} + y_{\min}}{2}$$

$$(origin)_z = 0$$

Sau đó, dịch chuyển mô hình sao cho gốc tọa độ nằm tại O bằng phép tịnh tiến:

$$T = \begin{bmatrix} -origin_x \\ -origin_y \\ -z_{\min} \end{bmatrix} \tag{4}$$

Mỗi điểm trong mô hình 3D được dịch chuyển:

$$P' = P + T \tag{5}$$

Với P=(x,y,z) là tọa độ ban đầu và P' là tọa độ sau tịnh tiến.

Thực hiện rời rạc hóa không gian miền chứa mô hình hình học thông qua xác lập khoảng cách lưới Δx, Δz. Các bước nhảy trên trục x và z xác định độ phân giải của tập hợp điểm. Độ phân giải cao (bước nhảy nhỏ) cho phép chi tiết hóa mô hình, nhưng tốn nhiều tài nguyên tính toán máy tính.

Phạm vi x_i và z_i được xác định:

$$x_i = x_{\min} + i \cdot \Delta x \tag{6}$$

$$z_j = z_{\min} + j \cdot \Delta z \tag{7}$$

$$i \in \{0, 1, 2, \dots, n_x\}, \quad n_x = \left\lfloor \frac{x_{\max} - x_{\min}}{\Delta x} \right\rfloor \tag{8}$$

$$j \in \{0, 1, 2, \dots, n_z\}, \quad n_z = \left\lfloor \frac{z_{\max} - z_{\min}}{\Delta z} \right\rfloor \tag{9}$$

Tiếp theo, thực hiện dò tìm giao điểm giữa tia và bề mặt bằng phương pháp Ray-Casting: Dò tìm các giao điểm giữa một tia (ray) xuất phát từ một điểm (x,z) trên mặt phẳng và hướng theo trục y. Dò tìm hai giao điểm: - y (điểm bề mặt mạn trái) và + y (điểm bề mặt mạn phải). Phương pháp này phù hợp để xử lý các mô hình phức tạp có nhiều chi tiết 3D.

* Dò tìm giao điểm bằng Ray-Casting:

- Mô hình tia dò:

Tại mỗi điểm lưới (x_i,x_i), một tia dò được tạo ra dọc theo trục y:

$$Ray_origin = (x_i, -\infty, z_j) \tag{10}$$

$$Ray_direction = (0, 1, 0) \tag{11}$$

- Phương trình tia:

Phương trình tham số của tia dò:

$$\vec{R}(t) = \vec{O} + t \cdot \vec{D} \tag{12}$$

Với:

$\vec{O} = (x_i, -\infty, z_j)$ - Điểm gốc của tia;

$\vec{D} = (0, 1, 0)$ - Hướng tia (trục y);

t - Tham số (khoảng cách dọc theo tia dò).

- Giao điểm tia và tam giác:

Mỗi tam giác trong mô hình 3D được biểu diễn bởi 3 đỉnh A, B,

C. Tìm t sao cho tia giao với tam giác:

$$\vec{R}(t) = A + u(B - A) + v(C - A) \tag{13}$$

Với: u, v ≥ 0, u + v ≤ 1, giải hệ phương trình (13) để tìm t, u, v.

t - Khoảng cách từ điểm gốc tia đến mặt tam giác;

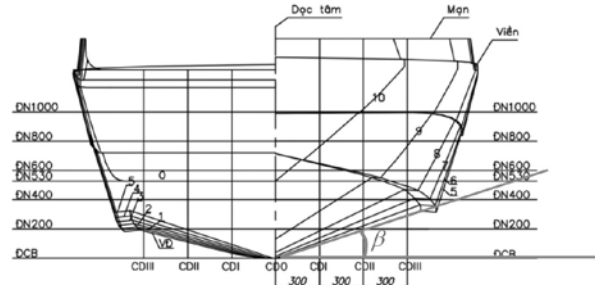
u, v - Tham số hình học trong tam giác.

Nếu tìm thấy t₁, t₂ thỏa mãn: t₁ < t₂ thì hai tọa độ y của giao điểm là: y = y(t₁), y* = y(t₂).

Xuất dữ liệu dạng point cloud: Dữ liệu được lưu trữ dưới dạng tệp tin *.CSV hoặc excel với cấu trúc cột [x y z], giúp dễ dàng phân tích hoặc tích hợp vào các công cụ khác như CAD, OpenFOAM hay phần mềm xử lý hình học.

3. KẾT QUẢ

Trong nghiên cứu này, chúng tôi dùng hình học tàu cỡ nhỏ chạy nhanh để rời rạc. Lý do lựa chọn này nhằm làm bước chuyển cho nghiên cứu tiếp sau về tối ưu hóa hình học tàu chạy nhanh thông qua chỉnh sửa tệp điểm hình học vỏ tàu (góc Beta) nhằm tăng hiệu suất thân tàu, cải thiện tính êm khi tàu di chuyển. Các cách làm quen thuộc như phương pháp Lackenby không hiệu quả do chiều chìm tàu chạy nhanh nhỏ, lượng chiếm nước khi đạt vận tốc lướt nhỏ và đặc biệt tàu có dạng gầy góc.



Hình 1. Hình học tàu nguyên mẫu (được trích từ dữ liệu hồ sơ thiết kế tàu cao tốc Ngân Hà 09 - Viện Nghiên cứu chế tạo tàu thủy, Trường Đại học Nha Trang, TP. Nha Trang, tỉnh Khánh Hòa)

Bằng phép quét tìm, chúng ta truy xuất được dữ liệu điểm rời rạc như minh họa trong Bảng 1. Các dữ liệu này đủ nhỏ để có thể nội suy bậc 1 hoặc bậc 2 mà không làm sai khác các giá trị trung gian, nếu dùng nội suy. Cách làm khác có thể chủ động đưa thông tin input ngay từ ban đầu rời rạc để nhận các giá trị x, y, z mong muốn nhằm phục vụ công tác phóng dạng, hạ liệu hoặc làm khuôn, dưỡng phục vụ công tác đóng tàu.

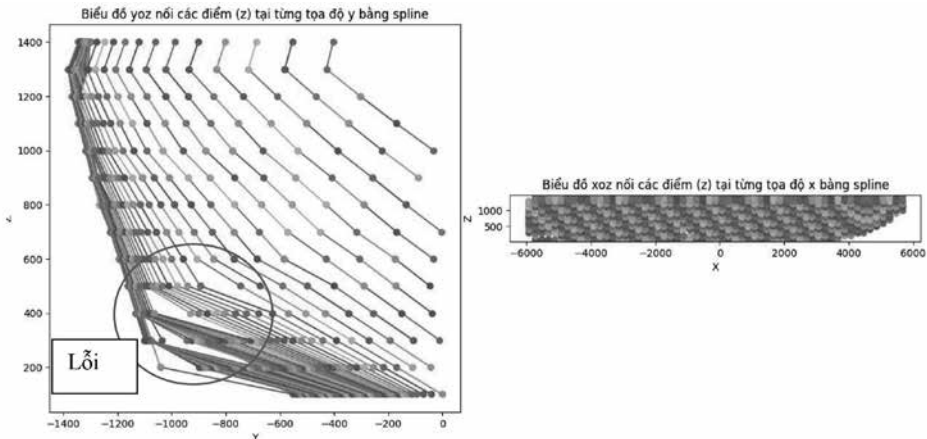
Bảng 1. Tệp dữ liệu điểm rời rạc

	A	B	C	D
1	x	y-	y+	z
2	-6135	-1226.64	1226.645	1110
3	-6135	-1238.54	1238.538	1120
4	-6135	-1248	1248.001	1130
5	-6135	-1256.24	1256.239	1140
6	-6135	-1263.68	1263.683	1150
7	-6135	-1270.55	1270.553	1160
8	-6135	-1275.8	1275.802	1170
9	-6135	-1275.81	1275.808	1180

Hình học tàu dạng gãy góc là loại đặc biệt thường dùng cho các tàu cỡ nhỏ chạy nhanh, rất khó thực hiện khi rời rạc và nội suy thông thường. Phương pháp quét tìm phát huy hiệu quả và khắc phục được nhược điểm trên, thực hiện được cho cả tuyến hình dạng U, V và U-V. Tuy nhiên, để mô tả tốt và đủ dữ liệu điểm cho các gãy góc, nên quan tâm kích thước các gãy góc trước khi xác lập kích thước ô dò tìm. Hình 2 minh họa cho trường hợp thông tin gãy góc

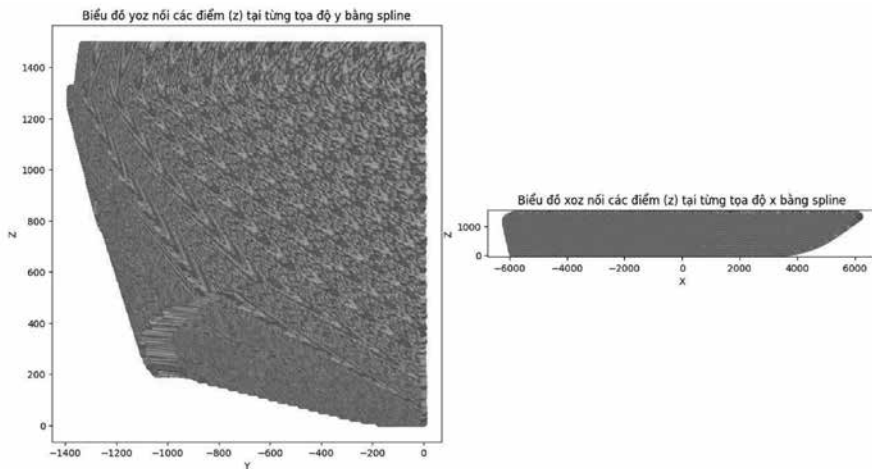
đủ diễn tả trong mặt sườn nhưng trong các mặt còn lại không đủ dữ liệu để biểu diễn. Khắc phục bằng cách tăng độ mịn cho ô quét tìm thông qua việc giảm kích thước bước tìm theo các phương trục x, y, z mong muốn.

Mô hình tàu ứng với các bước nhảy thô, không thể hiện được đặc trưng hình học vỏ tàu:



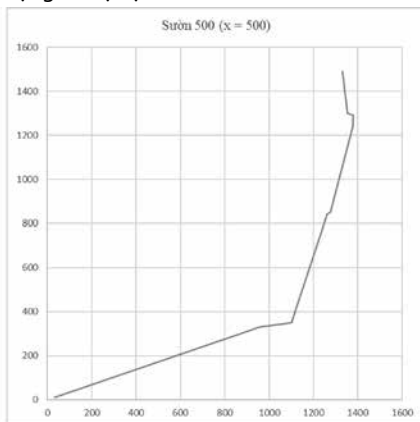
Hình 2. Hình học tàu rời rạc với bước lưới lớn

Mô hình tàu ứng với các bước nhảy tinh chỉnh, thể hiện được đặc trưng hình học vỏ tàu:

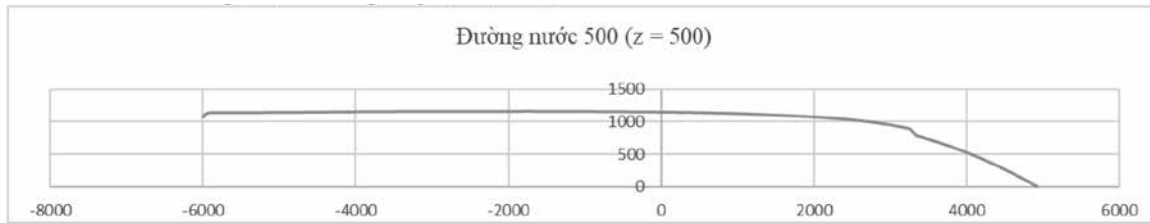


Hình 3. Hình học tàu rời rạc với bước lưới nhỏ (mịn)

Từ tập dữ liệu rời rạc, chúng ta có thể trích lượt hình học các sườn, đường nước bất kỳ, giúp thuận tiện trong công tác chế tạo khuôn hoặc gia công phông dạng và hạ liệu.



Hình 4. Biện dạng mặt cắt ngang tại tọa độ x = 500



Hình 5. Biên dạng mặt đường nước tại độ $z = 500$

Qua dữ liệu điểm, thể tích tàu cũng được tính toán bằng các thuật toán rời rạc, với diện tích đáy khối vi phân dV là $dx dy$.

4. THẢO LUẬN

Kết quả nghiên cứu đạt được là chương trình tự động hóa tính toán tệp dữ liệu điểm hình học vỏ bao tàu, ở đó các ô lưới tính toán xác lập thông qua đại lượng Δx , Δy , Δz do người dùng nhập, kỹ thuật điều chỉnh độ cong đột ngột biến đổi được xử lý trong hàm Spline bằng phương pháp thêm điểm lân cận với khoảng cách rất nhỏ nhằm luôn mô tả được các hình học cục bộ phức tạp như gẫy góc trên vỏ tàu, kết hợp kỹ thuật làm trơn. Dữ liệu điểm cho phép chuyển đổi sang định dạng lưới cấu trúc STL và dễ dàng tính toán thể tích chiếm nước V bằng tích phân các yếu tố thể tích $dV = z \cdot dx dy = A \cdot z$

$$V = \frac{1}{3} \sum_i A_i z_i \quad (14)$$

Tập dữ liệu điểm là thông tin đầu vào quan trọng cho các bài toán tối ưu điều chỉnh hình học vỏ tàu, đặc biệt các tàu chạy nhanh cỡ nhỏ hiện nay.

Hãy tưởng tượng rằng, mỗi một chu kỳ tính toán tối ưu sẽ nhận được 1 tệp tọa độ điểm và cần thiết mô hình hóa 3D tệp tọa độ điểm ấy để tính toán ứng xử về tính năng tàu, điều không thể làm thủ công đáp ứng được. Chương trình thật sự là công cụ đắc lực phục vụ cho công đoạn này.

Từ nghiên cứu này, một phát hiện mới về phương pháp tính toán thể tích tàu không đối ứng với tuyến hình tàu thay đổi được đề xuất thông qua ràng buộc trong bài toán tính thể tích V , sẽ được trình bày chi tiết ở bài báo sau. Phát hiện này giúp vòng lặp trong điều chỉnh hình học tàu giữ nguyên lượng chiếm nước được hình thành và nhiều mô hình tàu khác nhau được sinh ra, làm tài nguyên cho bài toán tối ưu tiến hành phân tích, đánh giá, cải tiến tính êm cho tàu cũng như các đặc tính thủy động lực học mong muốn.

Ngoài ra, kết quả của chương trình tự động hóa mà nhóm nghiên cứu còn ứng dụng cho công tác công nghệ phóng dạng và gia công hạ liệu, chế tạo khuôn, dưỡng chính xác.

5. KẾT LUẬN

Nghiên cứu đã xây dựng thành công chương trình tự động hóa rời rạc hình học vỏ tàu sang tọa độ điểm (x,y,z) và cấu trúc được dạng file.stl.

Nghiên cứu đã đề xuất được cách khắc phục biểu diễn spline với các điểm có độ cong cục bộ thay đổi lớn, kể cả tính chất đường hình gẫy góc bằng cách thêm điểm nội suy giữa kết hợp cách cắt đoạn hoặc nối tuyến tính tại điểm cong đột ngột.

Thêm điểm nội suy:

$$z_{new} = \frac{z_i + z_{i+1}}{2} \quad (15)$$

$$y_{new} = \frac{y_i + y_{i+1}}{2} \quad (16)$$

- Cắt và dùng đoạn thẳng để nối:

Tại điểm z_k có độ cong thay đổi đột ngột, sử dụng đoạn thẳng nối (y_k, x_k) và (y_{k+1}, z_{k+1}) thay vì spline.

TÀI LIỆU THAM KHẢO

- [1]. G. Kress (2011), Lofting: A Step-by-Step Guide for Boat Builders, 1st ed. Sheridan House.
- [2]. X. Zhang, Y. Fang and X. Chen (May, 2020), Development of Automated Tools in Ship Hull Form Processing, Journal of Ship Production and Design, vol.36, no.2, pp.145-154.
- [3]. A. Lee and B. Park (Jan, 2021), Point Cloud Processing in 3D Geometry Conversion for Marine Engineering Applications, Ocean Engineering, vol.230, p.108976.
- [4]. S. W. Smith (2020), Digital Signal Processing: A Practical Guide for Engineers and Scientists, 3rd ed. Elsevier.
- [5]. P. Liu, J. Gu and C. Wang (2021), Enhancing Intelligent Manufacturing in Shipbuilding via Cloud-Based Technologies, IEEE Access, vol.8, pp.9850-9865.
- [6]. P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli and G. Ranzuglia (2008), MeshLab: An Open-Source Mesh Processing Tool, in Eurographics Italian Chapter Conference, [Online]. Available: <https://www.meshlab.net/>.
- [7]. A. Jacobson, O. Baran, L. Kavan, J. Popović and O. Sorkine (Jul., 2012), Fast Automatic Skinning Transformations, ACM Transactions on Graphics (TOG), vol.31, no.4, pp.1-10, [Online]. Available: <https://igl.ethz.ch/projects/fast-winding-number/>.
- [8]. Trimesh Library, GitHub, 2023. [Online]. Available: <https://trimsh.org/>, Accessed: Dec. 26, 2024.
- [9]. G. Turk (1992), Re-Tiling Polygonal Surfaces, Computer Graphics Proceedings, Annual Conference Series, pp.55-64, doi: 10.1145/133994.134011.
- [10]. J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hughes (1995), Computer Graphics: Principles and Practice, 2nd ed. Addison-Wesley.
- [11]. F. Bernardini and H. Rushmeier (2002), The 3D Model Acquisition Pipeline, Computer Graphics Forum, vol.21, no.2, pp.149-172, doi: 10.1111/1467-8659.00574.
- [12]. R. J. Rost and B. L. Liceaga (2009), OpenGL Shading Language, 3rd ed. Addison-Wesley.
- [13]. Ray-Tracing Basics, NVIDIA Developer, 2023. [Online]. Available: <https://developer.nvidia.com/>, Accessed: Dec. 26, 2024.
- [14]. NumPy: Fundamental Package for Scientific Computing with Python, NumPy, 2024, [Online]. Available: <https://numpy.org/>, Accessed: Dec. 26, 2024.
- [15]. Pandas: Python Data Analysis Library, Pandas, 2024. [Online]. Available: <https://pandas.pydata.org/>, Accessed: Dec. 26, 2024.