PROPELLER SIMULATION IN OPEN-WATER CONDITION WITH BKASM - A USER INTERFACE BASED ON SNAPPYHEXMESH/OPENFOAM

MÔ PHỎNG CHÂN VỊT TÀU THỦY TRONG ĐIỀU KIỆN TỰ DO – GIAO DIỆN NGƯỜI DÙNG DỰA TRÊN CÔNG CỤ SNAPPYHEXMESH/OPENFOAM

Le Van Long, Mai Ngoc Luan, Nguyen Ho Nghia, Doan Minh Thien and Ngo Khanh Hieu
Department of Aerospace Engineering, Ho Chi Minh City University of Technology, VNU-HCM
longle3001@gmail.com, mngluan@gmail.com, ngokhanhhieu@hcmut.edu.vn

Abstract: *A propeller is a device converting rotary energy from the piston engines into the energy of the fluid thereby creating thrust to propel vessels. For this reason, it is extremely necessary to perform analysis on hydrodynamic characteristics of propellers and Computational Fluid Dynamics (CFD) can be considered the most sufficient way to accomplish this process. Nowadays, with the aid of the open-source software OpenFoam, researchers are allowed to carry out numerical simulations with high accuracy. To acquire this outcome, one must be aware of the importance of the meshing process. However, this procedure often requires skills and experiences from the conductors. Thus, in this paper, an automatic meshing tool utilizing snappy-HexMesh/OpenFOAM version 5.0 is introduced. As a user interface for snappyHexMesh, this tool assists researchers in open - water propeller simulations by simplifying the meshing process. The results obtained from the tool are validated by comparing with the experiment data provided by the manufacturer. From this, a conclusion can be made on the reliability of the meshing tool.*

Keywords: *OpenFOAM, snappyHexMesh, propeller, simulation…*

Classification number: *2.1*

Tóm tắt: *Chân vịt tàu thủy được áp dụng để tạo ra lực đẩy cho tàu bằng cách sử dụng năng lượng được truyền từ động cơ chính của tàu. Chính vì tầm quan trọng của chân vịt tàu thủy, các phân tích về đặc tính thủy động lực học cho nó trở nên rất cần thiết và phương pháp tính toán số động lực học lưu chất (CFD) được coi là giải pháp tiện lợi, ít tốn kém mà vẫn thể hiện đầy đủ và chính xác các đặc tính hoạt động của chân vịt. Ngày nay, với sự trợ giúp của phần mềm nguồn mở OpenFOAM, các nhà nghiên cứu có thể thực hiện các mô phỏng số với độ chính xác cao dựa trên các bộ tính toán có sẵn của phần mềm. Tuy nhiễn để có được một kết quả chính xác và đáng tin cậy, bên cạnh việc sử dụng bộ tính toán phù hợp, kỹ thuật chia lưới cho chân vịt cũng hết sức quan trọng và nó được xem là có ảnh hưởng trực tiếp đến kết quả của bài toán mô phỏng. Tuy nhiên, để tạo một lưới tốt cho vật thể, ta thường đòi hỏi phải có kỹ năng và kinh nghiệm trong lĩnh vực này. Vì lý do đó, trong bài báo này, nhóm tác giả muốn giới thiệu đến người dùng một công cụ chia lưới tự động dựa trên công cụ snappyHexMesh của phần mềm mã nguồn mở OpenFOAM phiên bản 5.0. Đây là một công cụ với giao diện người dùng trực quan sẽ hỗ trợ đắc lực cho các nhà nghiên cứu thực hiện các mô phỏng chân vị tàu thủy trong điều kiện tự do. Các kết quả thu được từ công cụ này được kiểm chứng bằng các so sánh với dữ liệu từ thực nghiệm của nhà sản xuất đưa ra. Từ đó, ta có thể kết luận mức độ phù hợp của lưới được chia tạo ra với bộ công cụ này với bài toán mô phỏng chân vịt trong điều kiện tự do.*

Từ khóa: *OpenFOAM, snappyHexMesh, chân vịt tàu thủy, mô phỏng số…*

Chỉ số phân loại: *2.1*

1. Introduction

There is a wide variety of publications related to propeller simulation with different objectives. The methods to construct a propeller mesh in these studies also varies and utilize a diverse number of meshing generators. To describe the motions in a rotating machinery problem, there are two popular methods, namely, Arbitrary Mesh Interface (AMI) and Multi Reference Frame (MRF). The former is often used with the PIMPLE solver for transient models and the latter goes with the SIMPLE solver for steady–state models [1]. In his study [2], Andreas Peters has chosen the AMI method to predict the effect of cavitation erosion on the P1225 propeller in oblique flow. Although he did not mention particularly what meshing tool he used, it can be observed that Peters contructed most of his computing field with hexahedra cells along with thin prism layers covering the boundary regions adjacent to the propellers. Similarly, in [3], Jianxi Yao also used the AMI method to analyse the hydrodynamic performance of a propeller in oblique flow. In his research, Yao employed a hybrid mesh, a combination of hexahedra cells in the outer part and tetrahedrons in the inner part with triangular prisms in the boundary layers. The author explained that the utilization of such hybrid mesh results from the complicated configuration of the propeller.

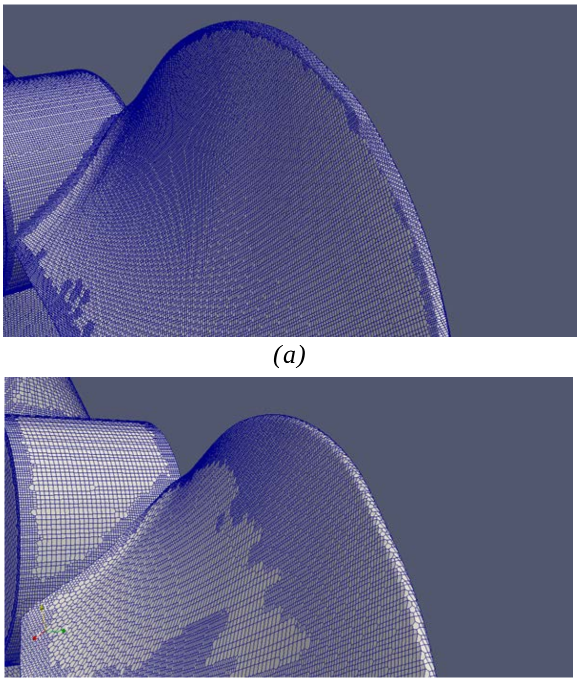
In steady – state cases, such as [4], Tuomas Turunen et.al applied MRF method and the SIMPLE solver to investigate a marine propeller in open – water conditions. The simulating grid they used includes hexahedrons generated by the snappyHexMesh utility together with prismatic cells in the layers and tetrahedrons in the propeller surface, which show a great similarity to [3]. The same as Turunen, Niroumand et.al also used the MRF method in his study on turbulent flow around the dtmb4119 propeller in open water [5]. However, the CFD mesh used in this publication comprises entirely of unstructured tetrahedrons.

Therefore, in this paper, a new approach to ship propeller simulation is introduced, which utilizes snappyHexMesh in the meshing process. So, the mesh is generated automatically by sHM and included of hexdominant cells. And the simulation of free - propeller in open water condition is based on the MRF approach with k - epsilon turbulence modelling. As a case study, the performance of a Wageningen B - series propeller obtained from simulation in comparison with the Wageningen B - series experiments has shown the reliability of our approach. And in conclusion, as a mesh generator integrated in OpenFOAM, snappyHexMesh holds a considerable economic advantage compared to other commercial meshing tools. This article would be a great support to researchers who are beginners in propeller simulation with OpenFOAM.

2. Propeller meshing approach with snappyHexMesh

2.1. Introduction to snappyHexMesh

SnappyHexMesh is an automatic meshing utility integrated in the open - source software OpenFOAM. This utility allows researchers to refine a coarse background mesh structure and turn it into a smoother one which meets the requirements of a numerical simulation problem. Follow this, before being able to use sHM, the conductors are requested to create a simple background mesh system. In OpenFOAM, another module named blockMesh is provided to perform this particular function [1]. Furthermore, there are various ways to construct a background mesh in snappyHexMesh cases, yet in this paper, the authors decide to use blockMesh for its suitability for propeller simulation problems. The mesh generated by sHM includes three - dimensional hexahedra and split-hexahedra cells. Besides, the geometries used in sHM cases are recommended to be in STL (stereolithography) format. With its automatic nature, snappyHexMesh's users are required to declare a number of initial parameters upon their demand for a CFD problem to create a complete and well-developed simulation mesh. From 2013 to 2015, snappyHexMesh has been developed relatively thorough for users in OpenFOAM version 2.0 series.



(a)

(b)

Figure 1. *Edge mesh generated by sHM in OpenFOAM: (a) version 2.1; (b) version 5.0.*

In 2015, OpenFOAM has been updated to the 3.0 version which comes along with multiple new mesh functionalities in order to improve the quality of the mesh generated by sHM, especially in cases with geometry made of curves and edges such as ship propellers (see fig. 1a and fig. 1b). This leads to an increase in studies employing sHM as an alternative meshing tool for other commercial options among the CFD community in the past 5 years. At the time of this study, OpenFOAM version 5.0 has been released with further improvements which not only promote the mesh quality but also enhance the usability of sHM.

To clarify the benefits of snappyHex-Mesh, the authors give a few comparisons between the mesh generated by sHM and the mesh constructed manually with another tool which utilizing tetrahedral cells. Table 1 shows some fundamental mesh quality parameters, table 2 shows the errors of the two sets of results and time required to reach the desired residual which is 5×10^{-5} for all cases. The mentioned two sets of mesh to be compared are created with a fairly similar number of cells. Also, the grid sizes and computational domains used which are established intentionally to provide the best mesh quality in both cases. The processing unit employed in this project is high performance with 6 cores and 12 threads along with 24GB of RAM.

As can be seen, the mesh from the commercial tool has a slightly better quality than that of sHM, but these differences do not reduce the reliability of the sHM mesh. On the other hand, the snappyHexMesh errors and time required are lower than those of the commercial mesh in almost every advance ratio. This partly proves the advantages of sHM in simulating open–water propellers.

Table 1. *Mesh quality comparisons between sHM mesh and commercial mesh.*

	sHM mesh	**Commercial mesh (Com mesh)**
Number of cells	2,871,231	2,778,491
Max non - orthogonality	65.2097	63.9777
Max Skewness	3.2409	2.47234
Max AR	21.9912	8.9122

In the numerical simulation of the dynamics of the propeller, the geometry of the object plays a decisive role in the final result. Using geometry in STL format with appropriate Deviation Tolerance (DT) – the maximum deviation between the original geometry and the STL presentation of that geometry, and Angle Tolerance (AT) – the value controls the maximum angle between the normal vectors of each triangle, will help improve mesh quality. In this paper, the authors choose the AT and DT parameters of the propeller based on the criteria for evaluating Max Skewness and Non-orthorgonality according to OpenFOAM standards.

Table 2. *Efficiency errors and numbers of iteration comparisons.*

	sHM mesh	Com mesh	sHM mesh	Com mesh
J	Δη (%)		Time (s)	
0.1	0.9373	8.546	3629	4041
0.2	1.4624	6.876	2855	3764
0.3	1.8506	5.294	2682	3231
0.4	2.2492	4.313	2875	2631
0.5	3.0463	4.482	2629	2463
0.6	5.5285	5.98	2546	1987
0.7	11.5675	2.24	2529	2428
0.75	0.0073	65.47	2418	2507

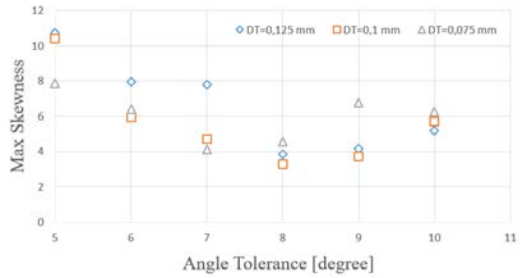


Figure 2. *Max skewness dependency on angle tolerance and deviation tolerance of geometry.*

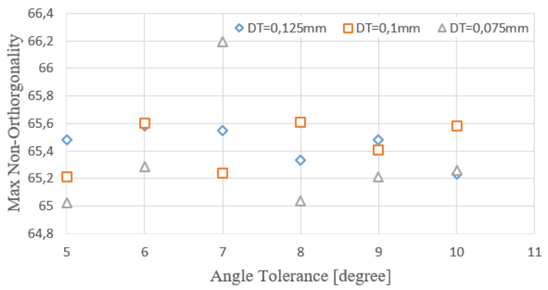


Figure 3. *Non-orthorgonality dependency on angle tolerance and deviation tolerance of geometry.*

According to the diagrams in figure 2 and figure 3, the parameters that match the STL format are DT of 0.1 and AT of 8 degrees. These settings are consistent with OpenFOAM mesh rating criteria.

2.2. Propeller simulation domains

On the basis of previously published papers [6][7], the authors continue to deve-lop the computational domain sizes which are more suitable for a sHM mesh system in order to maintain the balance among the number of cells, the total runtime required and the accuracy of the results. For this snappyHexMesh propeller case, the authors propose domain sizes as in Fig. 4. These sizes are adjusted to be smaller than those in [6][7], yet assure good mesh quality for accurate simulations.

Using the Multi Reference Frame (MRF) model, the rotating domain sizes have a great influence on simulation results. In this paper, the authors have conducted various test cases on different rotor domain sizes. The data acquired express best accuracy when the rotor region diameter and width are 1.15D and 0.3D, respectively (see Table 3). However, the width of this domain depends on the thickness of the propeller, this size must be sufficient to create boundary layers as well as not too far from the propeller surface, where in actually the fluid is less affected by the rotation of the propeller.

Table 3. *Test results of a number of sizes for the rotor (using the Wageningen B-series Propeller dimensions: A_e/A_0 = 0.45; P/D = 0.7; D=240 mm; 3 blades).*

Diameter	Width	% Ct	% Cq	% η
1.1D	0.25D	8.22	6.64	1.41
1.1D	0.30D	8.73	7.42	1.15
1.1D	0.35D	9.77	8.54	1.07
1.15D	0.25D	7.02	6.67	0.27
1.15D	0.30D	6.24	5.39	0.75
1.15D	0.35D	7.44	6.91	0.43
1.2D	0.30D	9.43	8.21	1.07
1.2D	0.35D	10.52	9.44	0.93
1.2D	0.45D	10.45	9.44	0.86

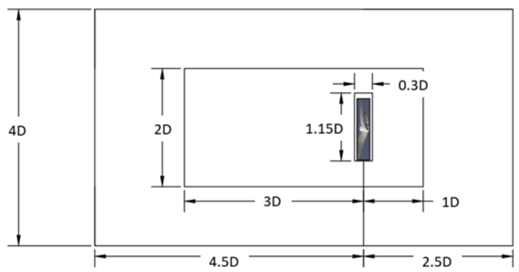


Figure 4. *Domain sizes.*

3. User interface and sample application

3.1. User interface

To take advantage of snappyHexMesh and to simplify the meshing process, the authors create a simple interface called BKAero SnappyMesher (BKASM). This tool is designed in order to support re-searchers in their early days of propeller simulations. Technically, BKASM is a user interface built on C++ foundation to provide aids in mesh generation particularly for open–water propeller cases (see fig. 5). This tool is integrated with sHM fundamental parameters, which have been modified and specialized for constructing a simulation mesh for propeller steady operation. Additionally, the k – ε turbulence model and the MRF method are the used to improve the key function of the tool, which is to find the optimal working range of marine propellers. Possessing this tool, users only need to import propeller geometry with recommended dimensions to create an applicable simulation mesh automatically.

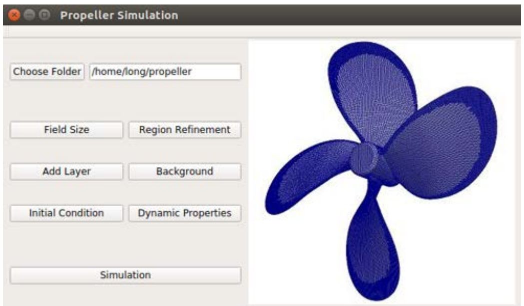


Figure 5. *BKASM main settings interface.*

In BKASM, we combined the three basic steps from preprocessing to processing in a sHM case. Users can adjust the settings for blockMesh background mesh in the Background button from the main interface or declare their own sHM parameters in the Field Size, Region Refinement, Add Layer settings and set up boundary condition values as well as field properties in the Initial Conditions and Dynamic Properties (see fig. 5). On the other hand, the parameters for the sHM step have already been included in the interface so users are able directly run the proposed numbers in the tool which can also provide good results (fig. 6, fig. 7, fig. 8, fig. 9).

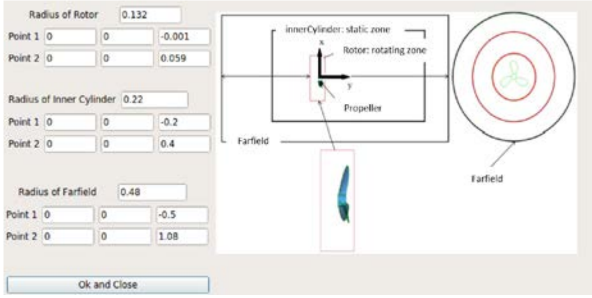


Figure 6. *Field size declarations interface.*

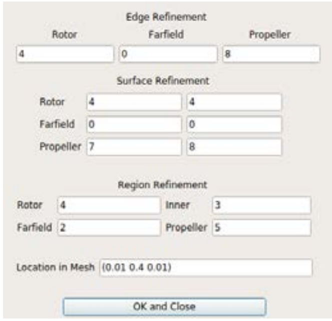


Figure 7. *Level of refinement interface.*

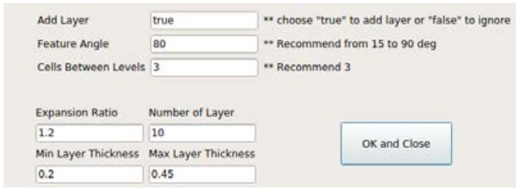


Figure 8. *Layer addition interface.*

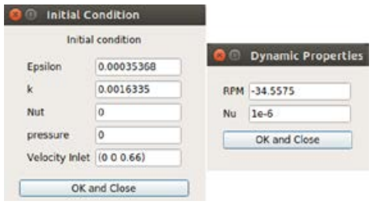


Figure 9. *Boundary conditions (left) and flow dynamic properties interface (right).*

3.2. Case study

To prove the feasibility of applying the mesh generated by BKASM into a propeller simulation case, the authors have done numerical analysis on different models of propeller. In this paper, the results of a three - blade Wageningen B - Series propeller are presented to examine the quality of the mesh system. Follow this, the thrust coefficient ($C_T = T/(\rho n^2 D^4)$), the torque coefficient ($C_Q = Q/(\rho n^2 D^5)$) and the efficiency ($\eta = C_T.J/(2\pi.C_Q)$) in the range of J from 0.1 to 0.75 are simulated and then compared to experimental data provided by the

Netherlands Ship Model Basin (N.S.M.B) at Wageningen [8].

The propeller used is scaled to a diameter of 240 mm (standard experimental diameter of Wageningen B - Series propellers [9]). The blade area ratio (A_e/A_0) and pitch ratio (P/D) at 0.7R are 0.45 and 0.7, respectively. The sHM parameters, the author proposed 10 boundary layers with the average value of y+ is 20 around the marine propeller. These parameters are consistent with the k-epsilon turbulence model and are proven to be quite good for computational simulations as shown in fig. 10. Computational domain sizes and boundary con-ditions are used as proposed in the tool. In general, the selection of the computational domain size as shown in fig. 4 and the boundary layer parameters as presented shows the mesh quality results as shown in table 4 and table 5.

Table 4. *Structure of the mesh components.*

Element Mesh Type	Cell Number
Hexahedral	2,230,646
Polyhedra	613,212
Prisms	27,012
Tet Wedges	361
Total	2,871,231

Table 5. *Evaluation of mesh generated by sHM.*

Evaluation Criteria	Value
Max non-orthogonality (<70)	65.2097
AR (<100)	21.9912
Max Skewness (<4)	3.2409
Layer Cover	97.2%

The non-dimensional coefficients are extracted from ParaView and displayed as performance characteristics in figure 10. The mesh created by snappyHexMesh/ BKASM has described quite exactly the Ct of this 3-blade propeller, with the errors are no greater than 6%. In Cq case, this value stands at 12%, which is identical to the Cp case. However, there is a slight increase in error at J = 0.75, this can be explained as one of the disadvantages of the k-ε turbulence model [10] (see Fig. 11). Nonetheless, the computed results have expressed accurately the optimal working range of the propeller compared to real–life situation, that is from J = 0.4 to J = 0.7. All things considered, the BKASM user interface has partially fulfilled its objective, that is to reduce the time and cost to conduct a simulation of a marine propeller for the beginners of marine propeller's CFD.

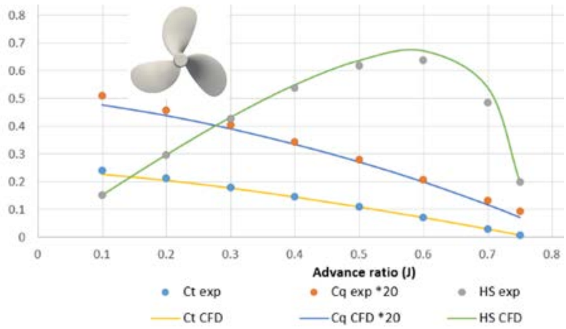


Figure 100. *Functional characteristic comparisons between numerical and experimental data in [8].*

4. Conclusion

This article has summarized some key points of mesh generation of snappy-HexMesh specialized for ship propeller simulation in open – water without the effect of water surface and accompanied objects. These points are included of suitable Deviation Tolerance and Angle Tolerance values of the STL geometry used in a sHM case providing the best mesh quality along with verified computing domain dimensions, sizes of mesh cells and thickness of boundary layers. Additionally, a user interface for sHM has been developed aims to help researchers who are not familiar with propeller simulation.

In future researches, authors continues to develop the sHM mesh with transient problems to describe the performance of the marine propellers under the influences of cavitation and multi - phase flow. These will be new features of the BKASM tool in later versions ❑

Acknowledgment

This research is funded by Ho Chi Minh City University of Technology, VNU-HCM, under grant number TNCS-KTGT-2017-09.

References

[1] Christopher J. Greebshields (2017). *User Guide OpenFOAM version 5*. OpenFOAM Foundation LTD.

[2] Andreas Peters et al., *Numerical Prediction of Cavitation Erosion on a Ship Propeller in Model and Full-Scale*, Wear, https://doi.org/10.1016/j.wear.2018.04.012.

[3] Jianxi Yao, *Investigation on hydrodynamic performance of a marine propeller in oblique flow by RANS computations*, Int. J. Nav. Archit. Ocean Eng. (2015) 7:56~69, http://dx.doi.org/10.1515/ijnaoe-2015-0005, pISSN: 2092-6782.

[4] Tuomas Turunen et al., *Open-water computaions of a marine propeller using OpenFOAM*, 11th World Congress on Computational Mechanics, Jul 2014.

[5] Amirhossein Niroumand et al., *Numerical simulation of turbulent flow around the dtmb4119 propeller in open water conditions*, Energy Equip. Sys, Vol. 6, No. 1, March 2018.

[6] P. Q. Thien, B. K. Huy, L. T. Hien, Ngo. K. Hieu, *Computational approach for a marine propeller based on Multi Reference Frame using OpenFOAM* , Journal of Transportation Science and Technology, Ho Chi Minh City University of Transport, Vol 20, Aug 2016.

[7] B. K. Huy, *Khao sat dac tinh luc day chan vit tau thuy cua tau song nho*, Aerospace Engineering Master Thesis, Ho Chi Minh City University of Technology, HCM-VNU, Jan 2016.

[8] M.M. Bernitsas et al., K_T, K_Q *and Efficiency Curves for the Wageningen B-Series Propellers*, Department of Naval Architecture and Marine Engineering, College of Engineering, The University of Michigan, 1981.

[9] Allan B.Murray et al., *Self – Propulsion Tests with Small Models*, Society of Naval Architects and Marine Engineers, Washington D.C, 1951.

[10] ANSYS, Inc, *Lecture 6 Turbulence Modeling*, Introduction to ANSYS FLUENT, Dec 2010.

Ngày nhận bài: 9/10/2018
Ngày chuyển phản biện: 12/10/2018
Ngày hoàn thành sửa bài: 2/11/2018
Ngày chấp nhận đăng: 9/11/2018