

OPTIMIZACIÓN DE FORMAS DE PROPULSORES NAVALES UTILIZANDO OPENFOAM

MARINE PROPELLER OPTIMIZATION USING OPENFOAM

Nicolás A. Antonelli^{a,b,c}, Gustavo E. Carr^{b,c} y Santiago A. Urquiza^{a,c}

^a*Grupo HidroSim, Universidad Tecnológica Nacional, Facultad Regional Mar del Plata, Buque Pesquero Dorrego N° 281, Mar del Plata, Argentina, email:hidrodinamica@mdp.utn.edu.ar*

^b*CONICET - Mar del Plata, Argentina, http://mardelplata-conicet.gob.ar/*

^c*Grupo de Ingeniería Asistida por Computadora (GIAC), Universidad Nacional de Mar del Plata, Facultad de Ingeniería, Juan B. Justo N° 4302, Mar del Plata, Argentina, email:ingenier@fi.mdp.edu.ar*

Palabras clave: OpenFOAM, Diseño naval, Propulsores navales, Hidrodinámica naval, Optimización de formas.

Resumen.

Para optimizar el rendimiento de hélices (propulsores navales) prediseñadas se puede hacer uso de técnicas de modelado computacional que comprendan la generación geométrica de las mismas, su mallado y cálculo de coeficientes de tiro y cupla en forma automatizada conjuntamente con algoritmos de optimización. En este trabajo se modela una hélice perteneciente a la serie B (serie de Troost) publicada por el canal de Wageningen, sobre la cual se realizan variaciones de parámetros geométricos. Se utiliza la plataforma GNU Salome (<https://www.salome-platform.org>) como entorno de trabajo para generar la geometría, vía scripts en lenguaje Python, y el mallador blockMesh perteneciente al paquete de código abierto OpenFOAM. Para resolver el flujo, se usa el método de volúmenes finitos a través del solver pimpleFoam (perteneciente a OpenFOAM) para realizar las simulaciones, mientras que para optimizar se implementa el método de Nelder-Mead de optimización sin restricciones, a través de la biblioteca SciPy. Como resultado se obtienen geometrías realistas que brindan mejores rendimientos propulsivos que las originales.

Keywords: OpenFOAM, Marine design, Marine propeller, Marine hydrodynamics, Shape optimization.

Abstract.

Computer-aided modeling techniques can be used to optimize the performance of prefabricated propellers (naval propellers), including their geometric generation, their meshing, and the calculation of traction and coupling coefficients along with optimization algorithms. In this work, a B-series propeller (Troost series) published by the Wageningen Channel is modeled, varying the geometric parameters. The platform GNU Salome (<https://www.salome-platform.org>) is used as a working environment to generate the geometry using scripts in the Python language and the BlockMesh mesher belonging to the open source package OpenFOAM. To solve the flow, the finite volume method with the solver pimpleFoam (part of OpenFOAM) is used to perform the simulations, while for optimization, the Nelder-Mead method is implemented for unconstrained optimization using the library SciPy. As a result, realistic geometries are obtained that provide better tunneling performance than the original geometries.

1. INTRODUCCIÓN

Las hélices son uno de los elementos unitarios más importantes del buque. Sus características determinan directamente el rendimiento del combustible a bordo, dado que su función es la de convertir la fuerza producida por el motor en fuerza axial para impulsar a la embarcación en el agua.

La información necesaria para definir el desempeño de una hélice se presenta en forma de diagrama con las curvas de coeficientes adimensionales de empuje y torque (KT , KQ) en función del coeficiente de avance (J). Para definir las características en condiciones de aguas abiertas, a menudo se recurre a pruebas experimentales con modelos a escala que giran y son arrastrados a velocidades conocidas. Sin embargo, existen limitaciones en la cuantificación del efecto de escala, lo que generalmente es un tema que no se comprende completamente. Esto disminuye la precisión de los procesos de extrapolación de modelos con diámetros pequeños y altas velocidades angulares a escala completa con diámetros mayores y bajas velocidades angulares (Carlton, 2019). Además, estas pruebas experimentales se consideran costosas y requieren mucho tiempo para obtener resultados.

Con el objetivo de la evaluación de parámetros de rendimiento en hélices, buques y artefactos navales, se está trabajando a nivel internacional en cuanto a experimentación de canales hidrodinámicos y su interrelación con los cálculos en laboratorios virtuales (métodos numéricos, dinámica de fluidos computacional). Tal es así, que actualmente se emplean técnicas de optimización computacional utilizando diseños paramétrica, generalmente con el objetivo de aumentar la eficiencia de los diseños, a la vez que se reducen los tiempos involucrados en el ciclo de diseño y, por tanto, la reducción significativa de costos de fabricación y de servicio.

En cuanto a la metodología para la resolución del problema de fludos, en Gaggero y Villa (2018) se analiza el rendimiento de una hélice marina con `pimpleFoam` validando la metodología, y en Bahatmaka et al. (2018) se comparan las estrategias AMI (Arbitrary Mesh Interface) y MRF (Moving Reference Frame). Por otra parte, en Sikirica et al. (2019) se estudia la influencia del modelo de turbulencia SST k - ω y K - ϵ en la predicción del rendimiento de hélices, donde hay que remarcar que actualmente el modelo más utilizado en estudios ingenieriles es el SST k - ω , aunque requiera un mayor refinamiento de malla.

En Vesting et al. (2016b) se utilizan un algoritmo genético de búsqueda no dominada elitista de segunda generación (NSGA-II en inglés de Nondominated Sorting Genetic Algorithm) y un algoritmo de optimización de emjambre de partículas (PSO, por sus siglas en inglés) para optimización multiobjetivo. En Vesting et al. (2016a) avanza en la misma línea utilizando una combinación del algoritmo PSO y NSGA-II para optimización multiobjetivo donde se presenta una metodología de tratamiento de la cavitación utilizando parámetros que controlan la convección de la pala como restricciones. En este artículo se plantean dos objetivos, por un lado la maximización del coeficiente de tiro (KT), y reducción de cavitación.

El objetivo del trabajo es evaluar el rendimiento y las características operativas de una hélice en aguas abiertas utilizando el software de código abierto OpenFOAM y desarrollar una metodología de optimización utilizando algoritmos existentes pertenecientes al paquete Scipy.

2. MÉTODO

Las ecuaciones que permiten modelar el comportamiento de los fluidos newtonianos se conocen ecuaciones de Navier-Stokes. Para tratarlas, se utiliza el método de volúmenes finitos para tratar las ecuaciones de Navier-Stokes promediadas por Reynolds (RANS) mediante el solver `pimpleFoam` perteneciente al paquete OpenFOAM.

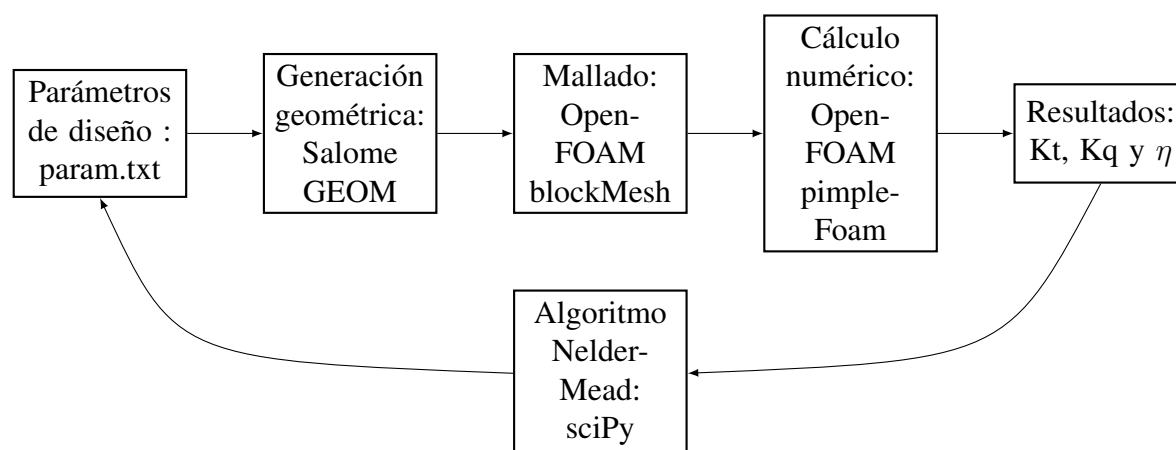


Figura 1: Esquema de optimización

Se utilizó el modelo $k-\omega$ SST para modelar la turbulencia y el método MRF (Moving Reference Frame, en inglés) para modelar la rotación de la hélice. Por otra parte, previo a la etapa de optimización, se realizó una validación de la metodología CFD, cuyos resultados arrojaron errores absolutos en torno a 0,5 % para k_t , 1 % para k_q y 2 % para el rendimiento, en comparación con los resultados experimentales (Kuiper, 1992).

Se plantea un esquema de optimización con restricciones utilizando el algoritmo Nelder-Mead (Nelder y Mead, 1965) mediante el paquete Scipy (Virtanen et al., 2020) representado por la figura 2.

2.1. Caso de estudio

El caso de estudio es una hélice perteneciente a la serie B de Wageningen de 4 palas y 1,5 metros de diámetro. (Kuiper, 1992).

Se parte del algoritmo de generación geométrica presentado en Carr et al. (2018) para construir la superficie de la hélice en función de una serie de parámetros principales, luego se utiliza el mallador blockMesh (perteneciente a OpenFOAM) y posteriormente se corre el caso "de fluidos" propiamente dicho, es decir, se resuelven las ecuaciones utilizando pimpleFoam. Los parámetros geométricos son:

- fa/F
- Diámetro
- Número de palas
- Paso
- Skew
- Rake

Los últimos dos son parámetros nuevos incorporados que se diferencian tanto de la descripción original de wageningen como del trabajo original (Carr et al., 2018).

2.2. Objetivos

Dentro de las funciones objetivo comúnmente utilizadas en este tipo de estudio se encuentran las relacionadas con: reducción del consumo de combustible, eliminación y/o reducción de vibraciones, incrementación del margen de seguridad frente a erosión por cavitación, maximización del rendimiento en velocidad, maximización del rendimiento en tiro, mejora de la integridad estructural, etc.

En la práctica se usan normalmente los coeficientes adimensionales K_t y K_q , para adimensionalizar el tiro y la cupla, respectivamente, según indican las ecuaciones 2 y 3. El primero se define a partir de la relación entre el empuje real de la hélice (llamado comúnmente "tiro") y el teórico; mientras que el segundo es la relación entre el par real (llamado típicamente "cupla") y el par teórico. El coeficiente de avance es el cociente entre la velocidad axial del flujo y la velocidad periférica (o de rotación) de la hélice, según indica la ecuación 1. Por otra parte, el rendimiento de la hélice se define como el cociente entre la potencia producida por la hélice y la potencia entregada a ella por medio del sistema propulsivo de la embarcación. No obstante, se puede expresar en función de los coeficientes K_t y K_q . En este trabajo, la función objetivo a maximizar será el rendimiento de la hélice.

$$J = \frac{V_a}{nD} \quad (1)$$

$$K_t = \frac{T}{\frac{1}{2}\rho n^2 D^4} \quad (2)$$

$$K_q = \frac{Q}{\frac{1}{2}\rho n^2 D^5} \quad (3)$$

$$\eta = \frac{K_t}{K_q} \frac{J}{2\pi} \quad (4)$$

2.3. Restricciones

El esquema contempla la restricción de los siguientes parámetros: diámetro, paso, relación de áreas f_a/F , número de palas y número de vueltas (velocidad de giro). Dentro del código de generación geométrica de la hélice, se fijan los parámetros geométricos mencionados para introducir dichas restricciones. Por otra parte, dentro del seteo del caso de OpenFOAM se impone la velocidad de rotación de la hélice como un valor constante.

3. RESULTADOS

Los resultados se describen en la figura 2. Se observa que se llega a la hélice óptima con ángulo de rake igual a $16,3^\circ$ y factor de skew igual a 1,34, para la cual se logra una mejora del rendimiento del 2 %

En la figura 3 y 4 se observan tanto la geometría original como la optimizada, donde puede observarse que la optimizada tiene mayor ángulo de Rake y Skew.

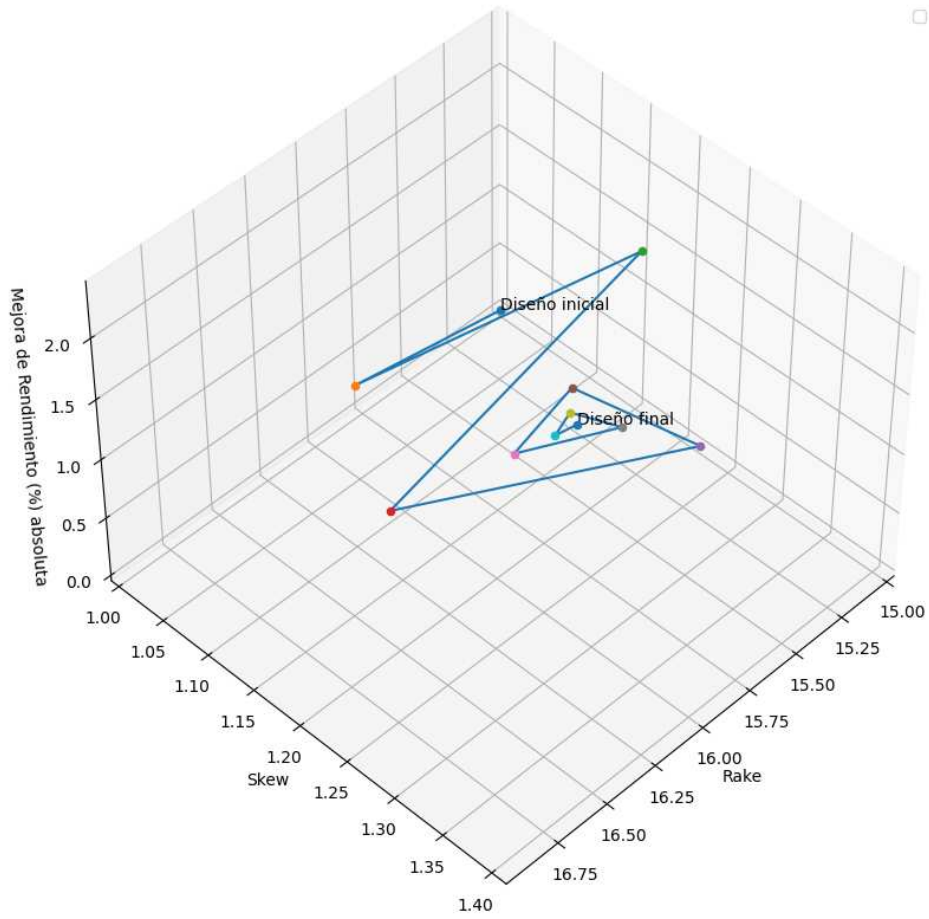


Figura 2: Resultados

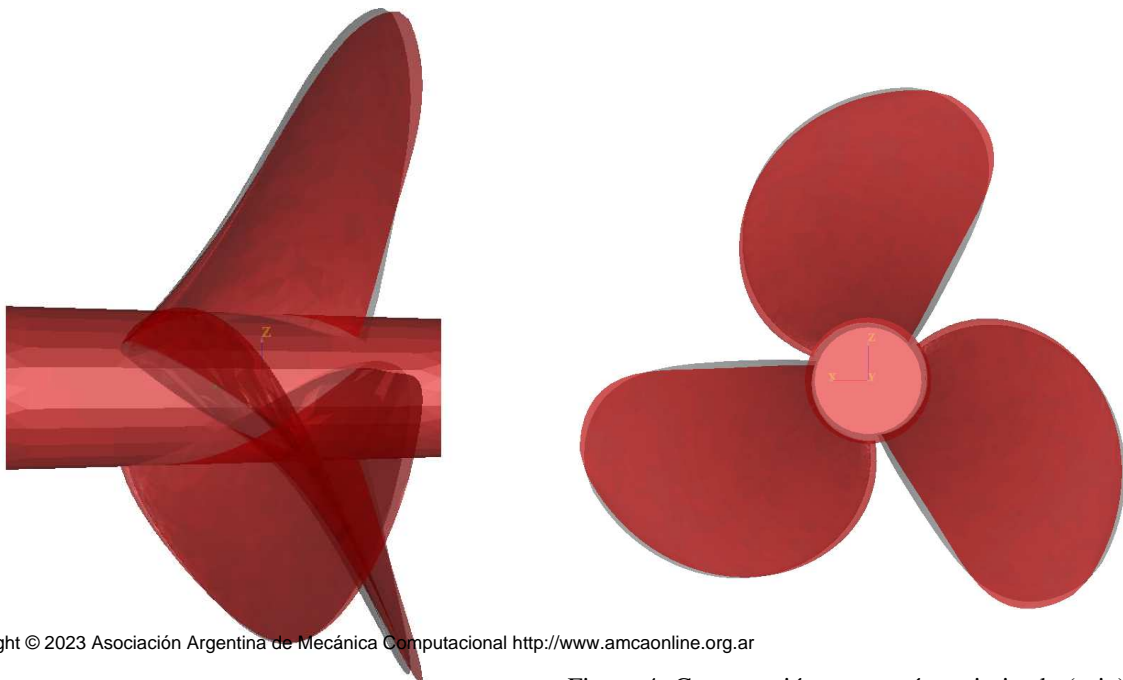


Figura 4: Comparación geometría optimizada (rojo) y geometría original (negro). Vista frontal

4. CONCLUSIONES Y TRABAJO FUTURO

Se parte de una geometría existente de hélice marina de amplia utilización en la industria naval, sobre la cual se realiza una parametrización geométrica introduciendo modificaciones en su geometría a través de 2 (dos) parámetros geométricos. Se realiza un ensayo de propulsor aislado mediante el método de volúmenes finitos sobre el cual se implementa un esquema de optimización.

El algoritmo de Nelder-Mead conjuntamente con la demuestra ser un método preciso, versátil y rápido para el diseño de hélices para optimización de una variable. Para el caso de estudio analizado, la geometría optimizada que maximiza el rendimiento resultó tener mayor rake y skew que la original; la mejora de rendimiento fue del orden del 2 %.

Si bien en el trabajo se presenta un esquema de optimización funcional se espera adaptar la técnica a los requerimientos específicos de diseño propios de cada tipo de embarcación (entre otras: rendimiento máximo en velocidad, rendimiento máximo en tiro, arrastre de redes de pesca), introduciendo tanto nuevas restricciones como nuevas funciones objetivo.

REFERENCIAS

- Bahatmaka A., Kim D.J., y Zhang Y. Verification of cfd method for meshing analysis on the propeller performance with openfoam. páginas 302–306. 2018. doi:10.1109/iCCECOME.2018.8659085.
- Carlton J. Chapter 8 - theoretical and analytical methods relating to propeller action. En J. Carlton, editor, *Marine Propellers and Propulsion (Fourth Edition)*, páginas 177–216. Butterworth-Heinemann, fourth edition edición, 2019. ISBN 978-0-08-100366-4. doi: <https://doi.org/10.1016/B978-0-08-100366-4.00008-0>.
- Carr G.E., Vaccari A., Giménez J.A., Sanchez Y., Biocca Nicolás Antonelli N.A., Martinez J.F., y Urquiza S. Automatización de generación de mallas para la optimización de propulsores navales. (artículo completo) *Mecánica Computacional. Industrial Applications*, XXXVI(12):499–510, 2018. ISSN 2591-3522.
- Gaggero S. y Villa D. Cavitating propeller performance in inclined shaft conditions with openfoam: Pptc 2015 test case. *Journal of Marine Science and Application*, 17, 2018. doi: 10.1007/s11804-018-0008-6.
- Kuiper G. *The Wageningen Propeller Series*. MARIN publication. Maritime Research Institute Netherlands, 1992. ISBN 9789090072470.
- Nelder J. y Mead R. A simplex method for function minimization. *Computer Journal*, 7:308, 1965. doi:<http://dx.doi.org/10.1093/comjnl/7.4.308>.
- Sikirica A., Čarija Z., Kranjčević L., y Lučin I. Grid type and turbulence model influence on propeller characteristics prediction. *Journal of Marine Science and Engineering*, 7(10), 2019. ISSN 2077-1312. doi:10.3390/jmse7100374.
- Vesting F., Bensow R., Johansson R., Gustafsson R., y Costa N. Procedure for application-oriented optimisation of marine propellers. *Journal of Marine Science and Engineering*, 4:83, 2016a. doi:10.3390/jmse4040083.
- Vesting F., Gustafsson R., y Bensow R.E. Development and application of optimisation algorithms for propeller design. *Ship Technology Research*, 63(1):50–69, 2016b. doi: 10.1080/09377255.2016.1145916.
- Virtanen P., Gommers R., Oliphant T.E., Haberland M., Reddy T., Cournapeau D., Burovski E., Peterson P., Weckesser W., Bright J., van der Walt S.J., Brett M., Wilson J., Millman K.J., Mayorov N., Nelson A.R.J., Jones E., Kern R., Larson E., Carey C.J., Polat İ., Feng

Y., Moore E.W., VanderPlas J., Laxalde D., Perktold J., Cimrman R., Henriksen I., Quintero E.A., Harris C.R., Archibald A.M., Ribeiro A.H., Pedregosa F., van Mulbregt P., y SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi:10.1038/s41592-019-0686-2.