

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/333704247>

Software de código abierto para un curso de Modelado, Simulación y Optimización de Sistemas Aeroespaciales

Conference Paper · September 2018

CITATION

1

READS

24

3 authors, including:



Yuriy Kovalenko

MSOLab; Cinvestav QRO

55 PUBLICATIONS 597 CITATIONS

[SEE PROFILE](#)



Yevgen Prokhorov

Center for Research and Advanced Studies of the National Polytechnic Institute

174 PUBLICATIONS 1,939 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Molecular relaxations [View project](#)



Synthesis of new materials [View project](#)

Tema A5. Educación en Ingeniería Mecánica: Software de código abierto

“Software de código abierto para un curso de Modelado, Simulación y Optimización de Sistemas Aeroespaciales”

Yuriy Kovalenko^{a*}, Roberto Ortega Escobedo^a, Yevgen Prokhorov^b

^aUniversidad Aeronáutica en Querétaro, Carretera Estatal 200, No. 22154, Parque Aeroespacial de Querétaro, Colón, Querétaro, C.P. 76278, México.

^bCentro de Investigación y de Estudios Avanzados del IPN - Unidad Querétaro, Libramiento Norponiente No. 2000, Fracc. Real de Juriquilla, Santiago de Querétaro, Querétaro, C.P. 76230, México.

*Dirección de correo electrónico: kovalenko.yuriy@gmail.com

RESUMEN

El artículo muestra un ejemplo de uso de software de código abierto en el curso de Modelado, simulación y optimización de sistemas aeroespaciales. En este ejemplo, se utilizaron los paquetes de modelador CAD en 3D - FreeCAD, de dinámica de fluidos computacional - OpenFOAM y el entorno de cálculo científico - Scilab para crear una plataforma de diseño óptimo de formas aerodinámicas. Se optimizó la forma de un cono truncado para obtener un coeficiente de arrastre mínimo al desplazarse en el aire. Los resultados del experimento numérico confirmaron la idoneidad de la plataforma desarrollada para encontrar la forma aerodinámica óptima. Adicionalmente, se muestra el uso de Scilab en el desarrollo del modelo dinámico de 6 grados de libertad de un vehículo aéreo y un modelo configurable 3D del sistema solar. Estos ejemplos comprueban la capacidad del uso de software de código abierto en la educación.

Palabras Clave: Software de código abierto, Modelado, Simulación, Optimización.

ABSTRACT

The article shows an example of open source software usage in the course of Modeling, simulation and optimization of aerospace systems. In this example, the 3D CAD modeler FreeCAD, the computational fluid dynamics package OpenFOAM, and the scientific software package Scilab were used to create an aerodynamic shape optimal design platform. The shape of a moving in the air truncated cone was optimized for a minimum drag coefficient. The results of the numerical experiment confirmed the ability of the developed platform to find the optimal aerodynamic shape. Additionally, the use of Scilab in the development of the 6 degrees of freedom dynamic model of an aerial vehicle and a configurable 3D model of the solar system, is shown. These examples prove the ability of using open source software in education.

Keywords: Open source software, Modeling, Simulation, Optimization.

1. Introducción

El desarrollo de un curso de modelado, simulación y optimización de sistemas aeroespaciales en México, surge por la necesidad de impulsar la creación de portales educativos para la educación en línea y la generación de contenido pertinente para estas áreas en particular, con el fin de fomentar la formación de capital humano y la divulgación de la ciencia, tal como se propone en el Programa Nacional de Actividades Espaciales de la Agencia Espacial Mexicana [1], y el Fondo Sectorial de Investigación, Desarrollo Tecnológico e Innovación en Actividades Espaciales del Consejo Nacional de Ciencia y Tecnología.

La importancia y relevancia del modelado computacional, radica en su utilidad, versatilidad y su amplio uso en la industria, ya que permiten una

experimentación libre de peligro y a bajo costo, lo cual representa una ventaja sustancial sobre la experimentación con objetos reales o modelos físicos, que frecuentemente suelen ser difíciles, peligrosos o inviables económicamente. Adicionalmente, los modelos son empleados con el fin de analizar, describir y poder simular fenómenos, procesos o sistemas, pues ayudan a comprender y predecir la respuesta del sistema simulado y, por ende, a identificar la oportunidad de mejora. En particular, el modelado y la simulación son especialmente importantes en la industria aeroespacial, en donde el método de ensayo y error es costoso y peligroso [2].

En este curso, es indispensable el uso de software computacional. En la actualidad, existen muchos paquetes computacionales que se emplean en la industria para el diseño asistido por computadora, el cálculo científico, la dinámica de fluidos, transferencia de calor, electromagnetismo, entre otras disciplinas. Sin embargo, estos paquetes suelen ser propietarios y requieren de

licencias de uso que por lo general son de renovación anual, y que implican un alto costo [3,4]. Por ello, con la intención de desarrollar este curso al menor costo, y hacer los resultados lo más accesibles para la comunidad, se propone utilizar principalmente herramientas de código abierto.

Al utilizar software de código abierto, implica que el código fuente de todo el producto sea accesible y modificable. También permite a los usuarios acceder o distribuir los programas desarrollados, sin pagar regalías. A su vez, los privilegios adjuntos al programa original se extienden a todos los que reciben el programa, por lo que los destinatarios no necesitan solicitar una licencia por separado. Más aún, el software de código abierto se puede usar en cualquier campo y para cualquier propósito, sin restricciones en la aplicación. Finalmente, puesto que no se emiten licencias sobre la base de la tecnología específica involucrada, implica que hay una neutralidad tecnológica.

Una vez desarrollado el curso, los materiales, simulaciones y optimización de sistemas no requieren de ningún mecanismo especial de transferencia, ya que todo estará disponible en línea en Internet.

El curso desarrollado consta de 3 secciones. La primera sección, "Modelado de objetos espaciales", abarca temas de cinemática de un proyectil, velocidades cósmicas (velocidades de escape), dinámica de un cohete con masa variable y resistencia del aire. En la segunda parte de la primera sección, las soluciones analíticas y numéricas del problema de dos cuerpos son discutidas. En la tercera parte de la primera sección, se discute la solución numérica del problema de n-cuerpos. La primera sección concluye con el desarrollo de un modelo configurable 3D del sistema solar. Como material auxiliar a la primera sección, se han desarrollado textos sobre integración numérica de sistemas de ecuaciones diferenciales y animación científica 3D en el software libre, de cálculo científico llamado Scilab [5]. Los resultados de la animación de las órbitas de los planetas del sistema solar se muestran en la Figura 1.

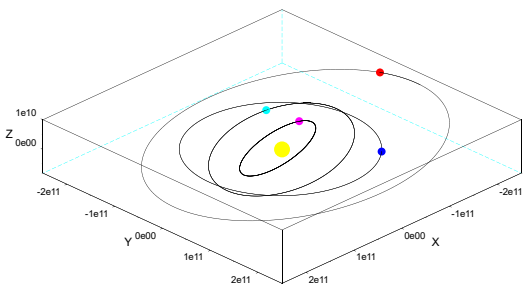


Figura 1- Resultados de la simulación de las órbitas de los planetas del sistema solar (planetas internos).

Una parte importante en la elaboración del curso consiste en la validación de los modelos realizados, lo cual se hace principalmente mediante la solución analítica,

como en el caso de cinemática de un proyectil, velocidades cósmicas, problema de 2 cuerpos, entre otros. Para aquellos modelos que no cuentan con solución analítica, la validación se hace al compararlos contra resultados conocidos o contra otros modelos. Tal es el caso del modelo del sistema solar desarrollado, cuyos datos obtenidos se compararon contra los datos del sistema solar JPL HORIZONS y del servicio de computo Ephemeris, el cual proporciona acceso a datos clave del sistema solar y a una producción flexible de efemérides de alta precisión para objetos del sistema solar como planetas, asteroides, cometas, entre otros. Estos sistemas son proporcionados por el *Solar System Dynamics Group* (Grupo de dinámica del sistema solar) del *Jet Propulsion Laboratory* (Laboratorio de propulsión a chorro) de la NASA (Administración Nacional de la Aeronáutica y del Espacio por sus siglas en inglés). Los resultados comparados comprenden las posiciones de 6 planetas (Tierra, Marte, Júpiter, Saturno, Urano y Neptuno) y sus respectivas distancias respecto al planeta Tierra (al Sol en el caso de la Tierra; Figura 2) al cabo de 50 años de trayectoria simulada. También se comparó contra eventos astronómicos determinados del año en curso, como la máxima proximidad entre la Tierra y Urano (23 de octubre), la máxima distancia entre la Tierra y el Sol (6 de julio), la mínima distancia entre la Tierra y Júpiter (9 mayo), entre otros. El error relativo máximo obtenido por el sistema fue de 0.18%

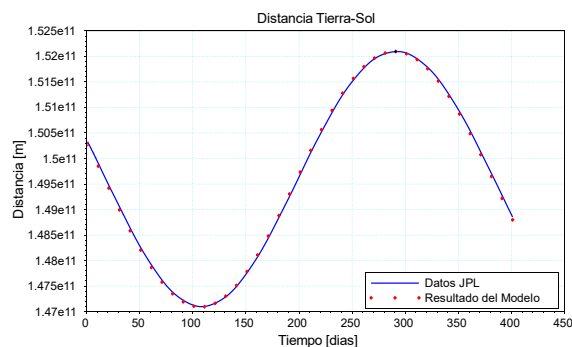


Figura 2- Distancia entre la Tierra y el Sol para los últimos 400 días de simulación.

En la segunda sección del curso, "Simulación de un vehículo aéreo de ala fija", se discuten los temas de sistemas de coordenadas, ángulos de Euler, así como la cinemática y dinámica de un vehículo aéreo. Estos temas sirven de base para el desarrollo del modelo dinámico de 6 grados de libertad de un vehículo aéreo con alas fijas, que también se presenta en esta sección. En la segunda parte de la segunda sección, los métodos de simulación de sensores de aceleración, giroscopios, sensores de presión y GPS son discutidos. La estimación del estado del vehículo por la inversión de modelos de los sensores, se presentan en la tercera parte de la segunda sección. Como material auxiliar a la segunda sección se han desarrollado textos sobre la

filtración de señales de los sensores, integración numérica de señales discretas y animación 3D de un vehículo aéreo en Scilab, la cual se muestra en la Figura 3.

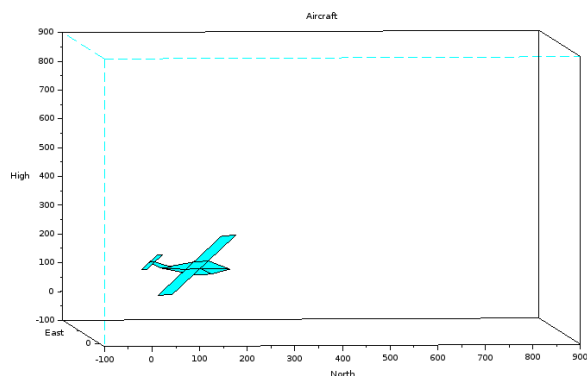


Figura 3- Animación de un vehículo aéreo en 3D en Scilab.

Para verificar el funcionamiento del simulador de la aeronave se realizaron comparaciones de las trayectorias recibidas por el modelo dinámico del avión con los resultados del algoritmo de estimación de orientación. El resultado mostró pequeñas diferencias entre el estimador y el modelo, cuyos errores se deben a la simulación del ruido de sensores, a la influencia del filtrado y a la integración numérica. El error máximo obtenido por el comparador fue de 0.020440. Este resultado se muestra en la Figura 4.

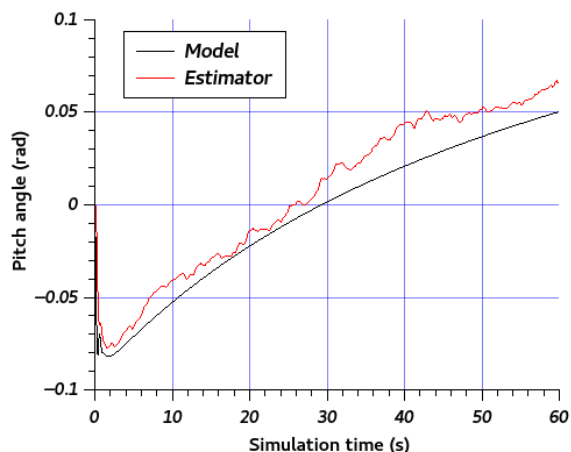


Figura 4- Cabeceo con respecto del tiempo con ruido y sin sesgo.

La tercera sección del curso ofrece una introducción al proceso de diseño óptimo de la forma aerodinámica de un objeto. El diseño óptimo consta de optimización multiobjetivo y multidisciplinario (por ejemplo, optimización aerodinámica en conjunto con optimización de estructura) [6]. Se resalta la diferencia entre el diseño convencional y el diseño óptimo y se dan las principales etapas del diseño óptimo. En la segunda parte de la tercera

sección, se proporciona un ejemplo de modelado y optimización de formas aerodinámicas con el uso de software de CFD (Dinámica de Fluidos Computacional por sus siglas en inglés) como un estimador. En este ejemplo, los estudiantes crearán una plataforma de optimización de forma aerodinámica usando solo software de código abierto. Esta plataforma de optimización es lo más simple posible, pero completamente funcional. La accesibilidad de software (gratuito, sin licencias y de código abierto) permite el desarrollo y uso de la plataforma incluso en las computadoras personales o portátiles de los estudiantes.

Los cursos enfocados al modelado y simulación de sistemas aeroespaciales [7] tradicionalmente usan software propietario, por ejemplo MATLAB. En este artículo, se muestra que dicho curso puede basarse exclusivamente en software libre. El resto del artículo expone un ejemplo del uso de software de código abierto para el diseño óptimo. El uso de software abierto permitió desarrollar una plataforma de optimización accesible que se describe a continuación.

2. Plataforma de optimización de formas aerodinámicas

La plataforma creada, está enfocada en optimización de geometrías aerodinámicas y vincula diversos paquetes computacionales. Por su diseño modular, permite emplear distintos algoritmos de optimización, y usa la simulación de la dinámica de fluidos para la estimación de aptitud de forma aerodinámica. El objetivo de optimización puede ser tanto el coeficiente de sustentación como el de arrastre.

2.1. Integración de paquetes computacionales de libre uso

La plataforma integra un software de diseño mecánico asistido por computadora - FreeCAD, un software de dinámica de fluidos - OpenFOAM, y un software de cálculo científico - Scilab.

Scilab es un entorno de cálculos y lenguaje de programación de alto nivel para cálculo científico, interactivo, de libre uso y disponible en múltiples sistemas operativos. Scilab es el software de integración de la plataforma, con el cual se realiza el vínculo entre los demás paquetes, y en el cual se realiza la optimización.

FreeCAD [8] es el software de diseño asistido por computadora que utiliza la plataforma para la generación y modificación de la geometría a optimizar. Este es un software multiplataforma que funciona como modelador 3D paramétrico, de código abierto y altamente personalizable y extensible, que puede fungir como extensión de otros programas.

OpenFOAM [9] (*Open - Field Operation and Manipulation*; Operación y manipulación abierta de campos) es un software de dinámica de fluidos computacional o CFD gratuito y de código abierto. Este es el software con el cual la plataforma estima la aptitud de la geometría del objeto a optimizar, por medio de simulación CFD.

Los paquetes anteriores tienen una interfaz de línea de comandos, que les permite ser utilizados en aplicaciones automáticas sin intervención humana, como lo es la optimización.

2.2. Estructura y funcionamiento de la plataforma

El diagrama de flujo del programa de optimización se representa en la Figura 5. El programa comienza con la inicialización del caso de OpenFOAM. El caso de OpenFOAM es una estructura de archivos, mostrada en la Figura 6, que contienen toda la información necesaria para correr una simulación CFD.

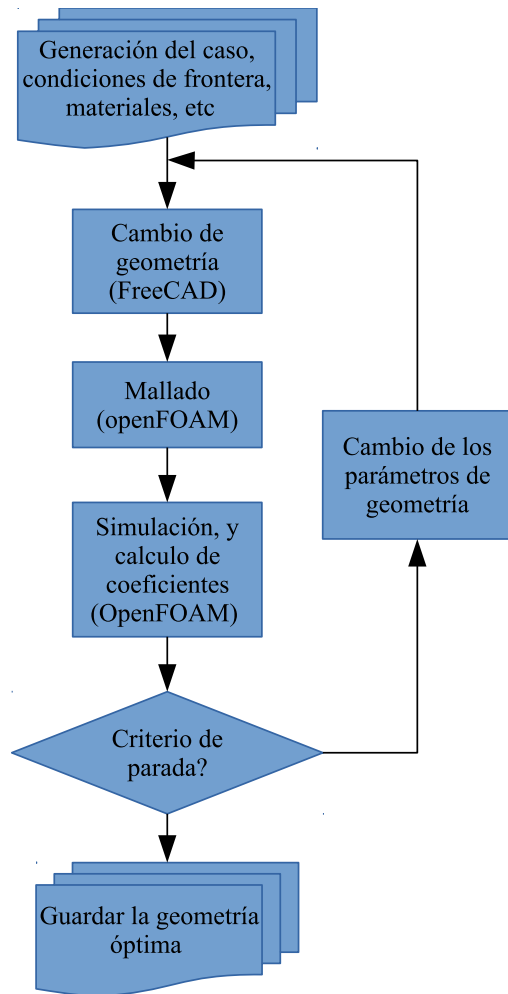


Figura 5- Diagrama de flujo del programa de optimización.

A continuación, el programa entra en el ciclo principal de optimización. En este ciclo, se realiza lo siguiente:

1. El programa genera nuevos valores para los parámetros de geometría a optimizar y produce un archivo de comandos en lenguaje Python, que es utilizado por el programa FreeCAD para crear la geometría del objeto de optimización. FreeCAD ejecuta la secuencia de comandos de Python para

generar y guardar en el disco un archivo de la geometría del objeto en formato STL (Lenguaje estándar de triángulos, por sus siglas en inglés).

2. El programa ejecuta las utilidades blockMesh y snappyHexMesh de OpenFOAM para crear y mallar el volumen circundante a la geometría del objeto, que fungirá como túnel de viento.
3. El programa usa el solucionador CFD llamado simpleFoam para resolver el caso.
4. El programa lee el archivo con los resultados del cálculo de los coeficientes aerodinámicos y toma el último valor del coeficiente de arrastre, cuyo valor es el objetivo de optimización.
5. El programa verifica la condición de parada. Si la condición de parada se cumple, el programa guarda la geometría óptima en el disco, de lo contrario, el programa repite el ciclo de optimización.

Actualmente, la plataforma utiliza dos diferentes algoritmos de optimización: algoritmo genético [10] y sección dorada [11]. El algoritmo de sección dorada es un algoritmo simple para optimizar funciones unimodales y unidimensionales. La ventaja principal de este algoritmo es la facilidad de implementación y su velocidad de trabajo. El algoritmo genético no tiene limitaciones en la dimensión del espacio de búsqueda y la forma de la función, sin embargo es complejo y lento.

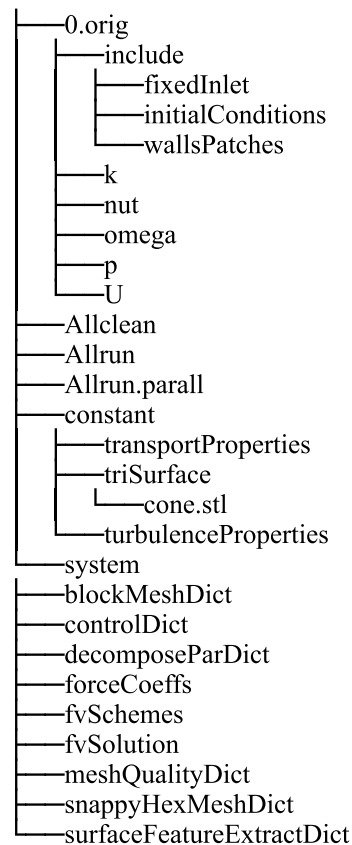


Figura 6- Estructura de archivos de OpenFOAM.

3. Experimento

Para probar el funcionamiento de la plataforma, se realizó un ejemplo de optimización de forma aerodinámica. En este ejemplo se muestra la solución del "Problema Aerodinámico de Newton" [12] utilizando el software de CFD. El "Problema Aerodinámico de Newton" consiste en encontrar el sólido de revolución que experimenta la menor resistencia al moverse con velocidad constante en un medio homogéneo disperso. Este problema resulta de gran utilidad, por ejemplo, en la construcción de torpedos, misiles o cohetes, en donde se busca una geometría en la cual se minimice la resistencia que opone el medio en el que se mueve el sólido. Se seleccionó este problema particular puesto que tiene una solución analítica para un cono truncado (Figura 7) en un medio disperso [12]. La solución de este problema se puede obtener minimizando la siguiente función:

$$f(x) = x^2 + (R^2 - x^2) \frac{(R - x)^2}{(R - x)^2 + H^2}, \text{ para } 0 \leq x \leq R$$

En donde R es el radio de la base del cono y H es la altura de cono trunco, como se muestra en la Figura 7.

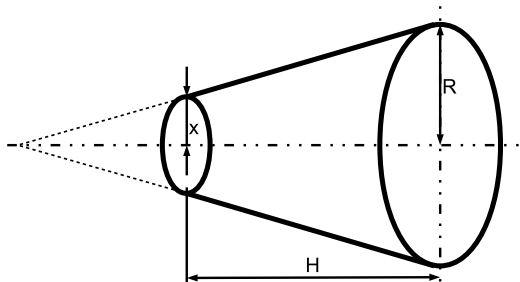


Figura 7- Cono truncado.

Para encontrar el mínimo de esta función se desarrolló un programa en Scilab que emplea el método de la sección dorada. Los valores del radio de la base del cono R y de la altura H son 1.5 m y 1 m respectivamente. La dependencia entre la resistencia del medio y el radio del cono truncado, se muestra en la Figura 8. El resultado del programa indica que el radio de la parte trunca debe ser igual a 0.779 m para formar la geometría que presenta menor resistencia. Es decir, el cono que genera menor resistencia al movimiento al desplazarse en el medio, es un cono trunco (no un cono con punta).

La solución de este problema de optimización con el uso de OpenFOAM como un estimador, se realizó en tres etapas.

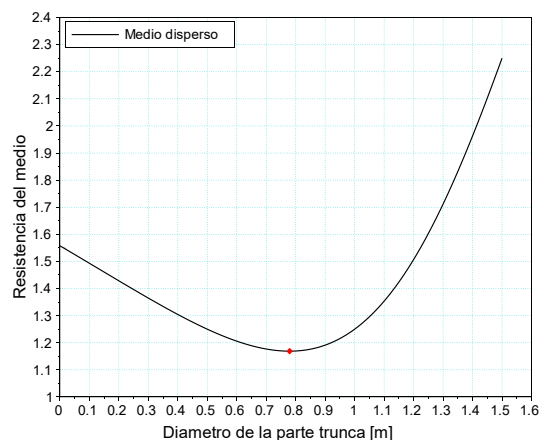


Figura 8- Dependencia entre la resistencia del medio y el radio del cono.

En la primera etapa se estableció un caso de OpenFOAM para la simulación CFD de flujo de aire (se tomó el aire como medio circundante) alrededor de un cono truncado. En particular, la geometría de un cono truncado en el formato STL se creó manualmente utilizando FreeCAD. Luego, en el archivo blockMeshDict, se establecieron los límites del área de simulación (la dimensión del túnel de viento virtual), así como también la entrada del flujo (inlet), la salida del flujo (outlet), y el conjunto de paredes (walls): frente, fondo, suelo y techo. Las fronteras se representan como se muestre en la Figura 9. Una vez definidas las fronteras y establecidas las condiciones iniciales y de frontera para el caso, se procede al mallado (discretización el dominio).

El comando blockMesh, es un generador de malla multi-bloque. Esto genera una malla, como se muestra en la Figura 10.

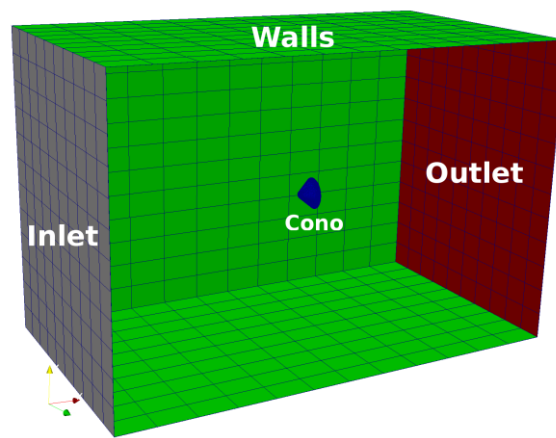


Figura 9- Fronteras.

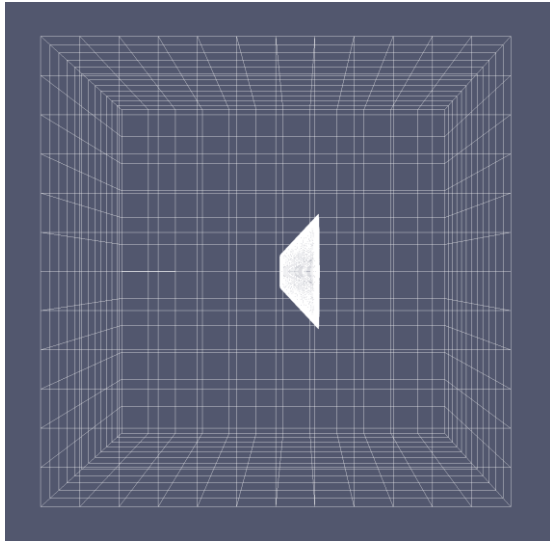


Figura 10- Discretización del volumen con malla multi-bloque.

Esta malla resulta muy burda, y por lo tanto se utiliza también el comando `snappyHexMesh`, el cual refina la malla y la ajusta a la superficie, para obtener una geometría más suave y llana, y con ello una discretización más apropiada del dominio, como se muestra en la Figura 11.

Después del mallado se procedió a simular el caso con el solucionador `simpleFoam`. Los resultados de la simulación CFD obtenidos son: distribución de presión sobre el cono (Figura 12), velocidad del aire alrededor del cono (Figura 13) y contornos de velocidad (Figura 14). El parámetro de interés para este caso, es la distribución de presión, pues está directamente relacionada con el coeficiente de arrastre.

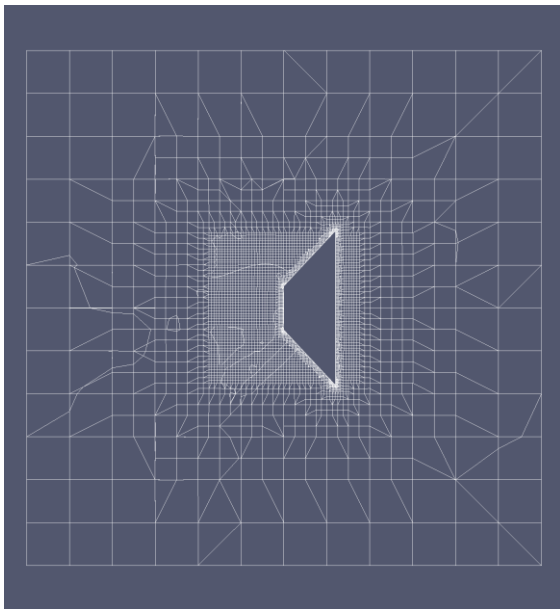


Figura 11- Discretización del volumen con malla refinada.

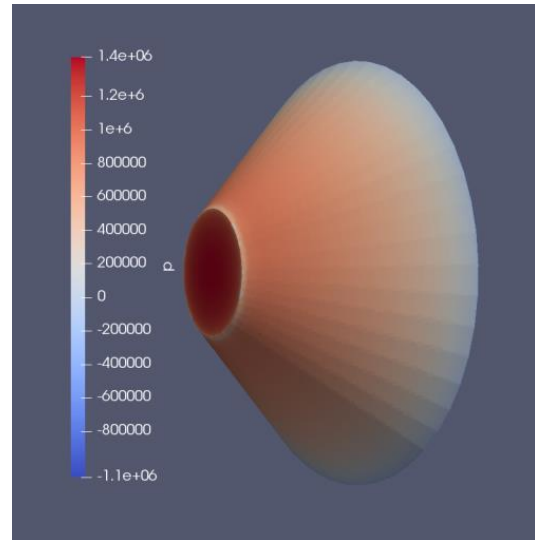


Figura 12- Distribución de presión sobre el cono.

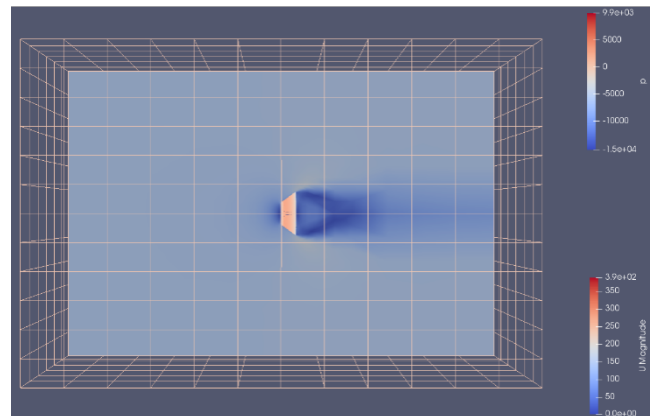


Figura 13- Distribución de presión sobre el cono y velocidad del aire alrededor del cono.

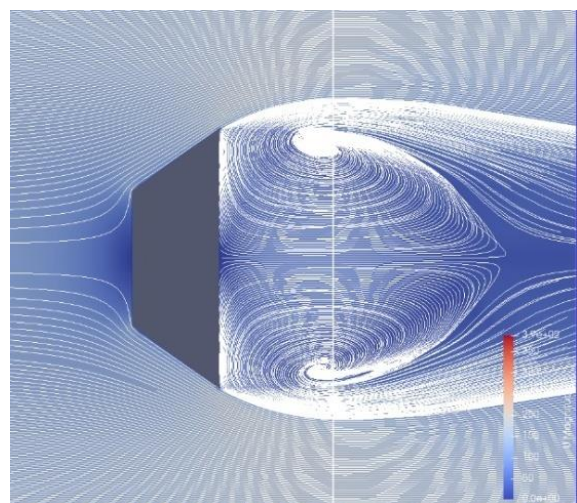


Figura 14- Contornos de flujo de aire alrededor del cono.

Como parte del post-procesamiento, se genera la siguiente tabla de coeficientes de momento, arrastre y sustentación.

Tabla 1: Coeficientes aerodinámicos obtenidos del post-proceso.

Iteración	Cm	Cd	Cl
1	-4.418776e-02	5.027251e-01	3.537532e-02
2	-2.697320e-02	5.034543e-01	2.779339e-02
3	2.887058e-02	4.701660e-01	-1.629548e-02
...
99	1.616227e-02	2.714984e-01	2.295311e-02
100	1.560038e-02	2.715541e-01	2.864916e-02

En la segunda etapa, se desarrolló un programa en Scilab que escanea el intervalo de búsqueda y modifica automáticamente el radio del cono con un cierto paso (desde 0, para formar un cono con punta, hasta el radio de la base, para formar un cilindro). En la Figura 15 se pueden ver tres de diez distintas variaciones de la geometría del cono trunco.

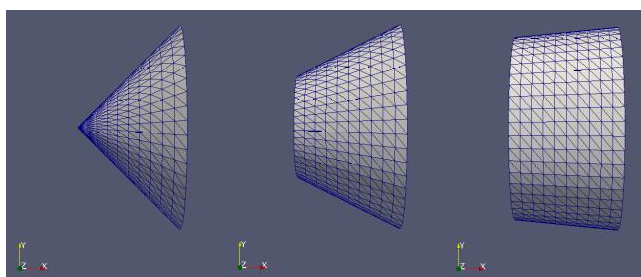


Figura 15- Distintas geometrías dentro del intervalo de búsqueda.

Posteriormente, el programa ejecuta la simulación de CFD y calcula el coeficiente de arrastre para cada una de las diez variaciones de la geometría. El resultado del escaneo del radio se muestra en la Figura 16.

Como se puede ver en la Figura 16, al variar el radio de la parte trunca del cono, también irá variando el coeficiente de arrastre. De las 10 geometrías generadas y simuladas, la que opone menor resistencia al flujo del aire al desplazarse a velocidad constante, es el cono con un radio de la parte trunca cercano a 1.1 m.

Los resultados recibidos en la segunda etapa distan del resultado analítico del problema aerodinámico de Newton (mínimo de la función mostrada en la Figura 8). Esto se debe a que el problema simulado y el problema Aerodinámico de Newton son esencialmente distintos, ya que este último considera choques perfectamente elásticos

entre las partículas del fluido y el sólido, y que las partículas están dispuestas de modo equidistante. En contraparte, el caso simulado contempla aire como medio circundante, el cual es compresible, turbulento y genera vórtices detrás del cono después del paso del fluido, tal como se muestra en la Figura 14. Estas diferencias afectan las condiciones de frontera del problema y con ello el resultado.

Finalmente, en la tercera etapa, se optimizó la forma aerodinámica del cono truncado utilizando la plataforma descrita en la Sección 2 del artículo. Inicialmente, se utilizó la función estándar del algoritmo genético de Scilab como algoritmo de optimización, sin embargo, el tiempo de ejecución de optimización excedió 24 horas. Por lo tanto, se realizó la optimización con el algoritmo de sección dorada, con el cual se obtuvieron resultados en menos de 1 hora de simulación.

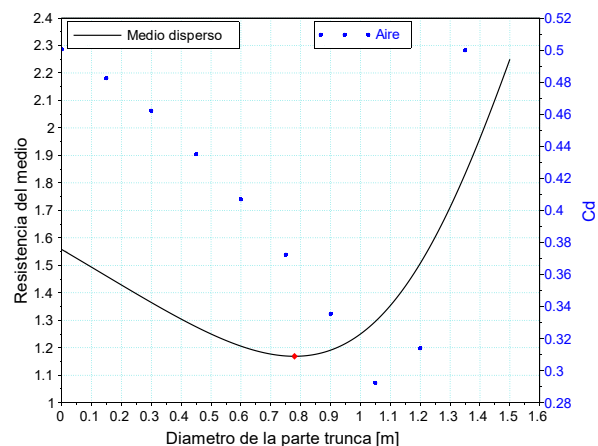


Figura 16- Coeficiente de arrastre vs radio de la parte trunca del cono.

En el proceso de optimización, el programa encontró en 20 iteraciones que el radio del cono que genera el mínimo arrastre es de 1.114 m, con una tolerancia de 0.015. Al comparar los resultados obtenidos de la segunda y tercera etapa del experimento, se puede concluir que la plataforma de optimización funciona correctamente.

4. Conclusión

El uso extensivo de modelos computacionales en la industria aeroespacial determina la demanda de un gran número de especialistas en este campo. La necesidad de formación de recursos humanos para el campo aeroespacial en México se observa en el Programa Nacional de Actividades Espaciales de la Agencia Espacial Mexicana. El Fondo Sectorial de Investigación, Desarrollo Tecnológico e Innovación en Actividades Espaciales del CONACYT, llama la atención sobre la falta de recursos

educativos en el tema aeroespacial. Tradicionalmente, los cursos de "Modelado y simulación de sistemas aeroespaciales" se basan en el uso de herramientas de desarrollo propietarias como Matlab, Ansys, Catia, etc. El curso presentado en este artículo está diseñado para uso exclusivo de software libre, lo cual hizo posible crear un curso más accesible y efectivo, sin las limitaciones inherentes al uso y manejo de licencias de productos propietarios.

El artículo ofrece varios ejemplos de modelado y visualización de resultados de simulación de sistemas dinámicos realizados por estudiantes en el marco del curso. Se muestra que el modelo 3D del sistema solar desarrollado es suficientemente preciso para fines educativos y presenta un error relativo del 0.18% respecto del modelo del JPL de la NASA.

Otro ejemplo de los programas desarrollados por los estudiantes durante el curso, es el modelo dinámico de 6 grados de libertad de una aeronave con alas fijas. La validación completa de este modelo sobrepasa los límites del curso educativo de 45 horas y de este artículo, por lo que el artículo solo compara las trayectorias recibidas por el modelo dinámico del avión con los resultados del algoritmo de estimación de orientación. El error encontrado en la estimación del cabeceo no excede 0.02044 radianes.

El ejemplo más complejo del uso de software abierto en los marcos del curso, es la plataforma de optimización de forma aerodinámica. Esta plataforma integra tres paquetes computacionales: FreeCAD, OpenFOAM y Scilab. El artículo da un ejemplo de optimización de un cono truncado para obtener una geometría con una resistencia mínima al aire. Se muestra que la solución encontrada con el uso de OpenFOAM no concuerda con la solución analítica conocida de este problema, ya que la solución analítica no tiene en cuenta las propiedades físicas del medio real. Para probar la eficacia funcional de la plataforma de optimización, se analizó la dependencia del coeficiente de arrastre con el radio del cono truncado. Los resultados del escaneo del radio y la optimización coincidieron bien, lo que confirma la capacidad de funcionamiento de la plataforma desarrollada.

Actualmente, el curso está siendo probado en la Universidad Aeronáutica en Querétaro (UNAQ.) Al final de las pruebas todos los materiales del curso, incluyendo códigos fuente de los programas, estará disponible en un sitio web en forma gratuita.

Agradecimientos

Proyecto apoyado por el Fondo Sectorial de Investigación, Desarrollo Tecnológico e Innovación en Actividades Espaciales.

REFERENCIAS

- [1] Agencia Espacial Mexicana, Programa Nacional de Actividades Espaciales 2013-2018. Ciudad de México, Distrito Federal (2016).
- [2] Fasano, G., & Pintér, J. D. (Eds.). (2012). *Modeling and optimization in space engineering* (Vol. 73). Springer Science & Business Media.
- [3] Lakhan, S. E., & Jhunjunwala, K. (2008). Open source software in education. *Educause Quarterly*, 31(2), 32.
- [4] Pires, P. S. M., & Rogers, D. A. (2002). Free/open source software: An alternative for engineering students. In *Frontiers in Education, 2002. FIE 2002. 32nd Annual* (Vol. 1, pp. T3G-T3G). IEEE.
- [5] Scilab Enterprises (2012). Scilab: Free and Open Source software for numerical computation. Available from: <http://www.scilab.org>
- [6] Skinner, S. N., & Zare-Behtash, H. (2017). State-of-the-art in aerodynamic shape optimisation methods. *Applied Soft Computing*.
- [7] Tewari, A. (2007). *Atmospheric and space flight dynamics*. Birkhäuser Boston.
- [8] Juergen Riegel, Werner Mayer, Yorik van Havre (2001-2017). FreeCAD (Version 0.16.6712) [Software] Available from <http://www.freecadweb.org>
- [9] OpenCFD Ltd (ESI Group), (2017) [Software], Available from <https://www.openfoam.com>
- [10] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197.
- [11] Mathews, J. H., & Fink, K. D. (2004). *Numerical methods using MATLAB* (Ed. 4). Upper Saddle River, NJ: Pearson Prentice Hall.
- [12] Tikhomirov, V. M. (1990). *Stories about maxima and minima*. Universities Press.