UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TRABAJO DE FIN DE GRADO

Grado en Ingeniería Mecánica

SIMULADOR PARAMÉTRICO DE VUELO PARABÓLICO

Volumen I

Memoria

Autor: Ignacio Fernández Ciérvide
Director: Miguel Ángel Brigos Hermida
Codirector: Antoni Pérez-Poch
Departamento: EGE – Departamento de Expresión Gráfica en la Ingeniería

Convocatoria: Enero 2020

Resumen

Este proyecto es la continuación a una serie de trabajos en los cuales, mediante el software de CAD, SolidWorks®, se han simulado vuelos parabólicos con diferentes aeronaves respectivamente. El objetivo es crear un simulador paramétrico que permita aunar a todos estos proyectos en uno, es decir, que desde una única aplicación se puedan introducir los datos variables según la aeronave deseada y ejecutar el estudio.

Para ello se ha llevado a cabo un proceso de investigación en la documentación, proporcionada por los creadores de SolidWorks®, sobre la interfaz de programación de la propia aplicación. Mediante las subrutinas, funciones y métodos disponibles, con el lenguaje de programación Visual Basic .NET, se ha creado un complemento para el software SolidWorks® con su propio instalador con el que el usuario podrá: ejecutar automáticamente la simulación de movimiento del vuelo parabólico con el complemento *Motion*, ya presente en SolidWorks®, y descargar un informe en Excel con los resultados de salida.

De esta manera se podrá conocer fácilmente la altura que alcanza la aeronave en el vuelo y la fuerza G a la que está sometida en cada momento.

Resum

Aquest projecte es la continuació d'una sèrie de treballs en els quals, amb el software de CAD, SolidWorks®, s'han simulat vols paramètrics amb diferents aeronaus. L'objectiu ès crear un simulador que permeti englobar tots els treballs en un, es a dir, que des d'una mateixa aplicació es pugui introduir les dades variables segons l'aeronau desitjada i executar l'estudi.

Per tal de fer-ho possible, s'ha fet un procés d'investigació en la documentació, proporcionada pels creadors de SolidWorks®, sobre la interfície de programació de la pròpia aplicació. Mitjançant les subrutines, funcions y mètodes disponibles, amb el llenguatge de programació Visual Basic NET, s'ha creat un complement per al software SolidWorks® amb el seu propi instal·lador amb el que l'usuari podrà: executar automàticament la simulació de moviment de vol parabòlic amb el complement *Motion*, ja present en SolidWorks®, i descarregar un informe en Excel amb els resultats de sortida.

D'aquesta manera es podrà conèixer fàcilment la altura a la que arriba la aeronau en el vol i la força G a la que està sotmesa en cada moment.

Abstract

This project is the continuation of a series of studies in which, by means of CAD software, SolidWorks®, parabolic flights are simulated with different aircraft respectively. The goal is to create a parametric simulator that allows all these projects to be combined into one, that is, that from a single application the variable data can be entered according to the desired aircraft and run the study.

For this purpose, a research process has been carried out in the documentation, provided by the creators of SolidWorks®, on the programming interface of the application itself. Through the subroutines, functions and methods available, with the programming language Visual Basic .NET, an add-in has been created for the SolidWorks® software with its own installer with which the user can: automatically run the simulation of parabolic flight movement with the Motion add-in, already present in SolidWorks®, and download a report in Excel with the output results.

In this way, it will be easy to know the height of the aircraft on the flight and the G force to which it is subjected at all times.

Agradecimientos

A mi familia, en especial a mis padres, que probablemente sin ellos no hubiera llegado hasta aquí.

A Ana, que ha sido gran parte de mi motor en esta última etapa universitaria.

A mi tutor, que me ha dado la oportunidad de aprender más de lo aquí se va a mostrar.

A toda la gente que ha estado cerca durante estos días de universidad.

Glosario

CAD (Computer-Aided Design): técnica de diseño basada en el uso de ordenadores (o estaciones de trabajo) como ayuda en la creación, modificación, análisis u optimización de un diseño.

Interfaz: límite a través del cual dos o más componentes separados de un sistema informático intercambian información. En la programación orientada a objetos, que es como se considera aquí, es un tipo abstracto que no contiene datos, pero define comportamientos de los parámetros de entrada

API (Application Programming Interface): conjunto de subrutinas, funciones y métodos que ofrece una biblioteca para ser utilizada por otro software como capa de abstracción.

VBA (Visual Basic for Applications): lenguaje de programación basado el Visual Basic, que permite crear funciones definidas por el usuario (UDFs), automatizar procesos y utilizar otras funcionalidades de bajo nivel a través de las bibliotecas de enlace dinámico (DLL). Actualmente integrado en las aplicaciones de Microsoft Office, SolidWorks® y AutoCAD, entre otras.

VB.NET (Visual Basic.NET): lenguaje de programación completamente orientado a objetos, basado en Visual Basic e implementado en el entorno de trabajo de Microsoft .NET. Es incompatible con VBA.

Macro: serie de instrucciones creadas para automatizar tareas en el entorno de una aplicación. Las herramientas para crear, editar y guardar pueden estar proporcionadas por la propia aplicación.

Add-in application (Complemento): es un componente software que se añade para proporcionar una prestación a un programa ya existente. (También se le conoce como plug-in o add-on)

Stand-alone application (Aplicación independiente): aplicación que no requiere ser instalada y que puede funcionar de forma totalmente independiente con respecto a otro software. Es decir, que, sin ser una extensión del software, puede conectarse a él cuando es requerido.

***.swp**: extensión de un macro procedente del software SOLIDWORKS®

***.exe**: extensión de un archivo que se puede ejecutar

***.dll**: extensión de un archivo con código ejecutable. Se ejecutan bajo demanda de un programa por parte de un sistema operativo.

***.vb**: extensión de un archivo escrito en lenguaje Visual Basic

***.msi**: componente de Microsoft que permite la instalación, mantenimiento y eliminación de un software

Class (Clase): es una plantilla para la creación de objetos de datos, proporcionando propiedades y funciones. Cada objeto creado a partir de la clase se denomina instancia de clase.

Object (Objeto): puede ser una variable, una estructura de datos, una función o un método. Según la programación orientada a objetos, es una instancia particular de una clase.

Método: son las acciones que puede realizar un objeto. Guardan similitud con las funciones matemáticas, aunque no tienen por qué aceptar parámetros.

Propiedad: se utilizan para especificar y devolver el estado de un objeto, es decir, lo describen. También se les denomina atributos.

COM (Component Object Model): interfaz binaria que permite la utilización de objetos en un entorno diferente al que han sido creados. La conversión de los tipos de objetos entre diferentes interfaces se consigue aplicando el método QueryInterface().

QueryInterface(): método de la interfaz IUnknown que permite al que hace la llamada recuperar referencias a las interfaces que implementa el componente. Se utiliza para obtener una manera de señalar a otra interfaz, dado un GUID que la identifica de forma exclusiva.

GUID (Globally Unique Identifier): es un código de 36 caracteres (32 dígitos y 4 guiones) que se utiliza para otorgar una identificación única a un objeto en diferentes contextos. En este caso se utiliza para identificar un tipo interfaz.

Call-back (Devolución de llamada): las call-backs de llamada son funciones dentro de su programa que se llaman desde otro proceso bajo ciertos eventos.

Índice

1.Introducción

1.1. Origen del trabajo

Los inicios de esta línea de investigación se remontan al año 2014, con la publicación de un artículo [1] por Miguel Brigos, Antoni Pérez-Poch y Francisco Alpiste, en el que se reportan los resultados de la aceleración residual obtenida en varios vuelos parabólicos de prueba y se propone un método de simulación, con el software de CAD SOLIDWORKS®, para optimizar el trazado de la parábola. Así, se consigue evitar pruebas previas al vuelo definitivo y, por tanto, más rentable.

Una serie de trabajos de fin de grado han seguido esta línea con el fin de mejorar los resultados, ya sea probando otras aeronaves o bien modificando la configuración del simulador.

1.2. Motivación

En primera instancia, los vuelos parabólicos para aproximarse a la gravedad nula se hacen con el fin de aclarar dudas sobre cómo puede afectar la ausencia de gravedad a las infraestructuras y a la vida en el espacio. Además de que se trata de un método más económico.

Pero este proyecto, en concreto, pretende simplificar los pasos que debe dar el investigador durante el uso de la herramienta de optimización del trazado del vuelo parabólico, que no es otra que el simulador creado con SOLIDWORKS®. Procurando, así, invertir menos tiempo en procedimientos mecánicos, siendo estos automatizados.

1.3. Requerimientos previos

Para comenzar, es necesario tener en cuenta que el autor ha tomado como referencia un trabajo de final de grado concreto, el de Francesc Llobera (2017) [2]. Por tanto, a pesar de posibles mejoras que haya habido en futuros estudios, solo se han tenido en cuenta las características de simulación de este.

Los parámetros escogidos dentro del software, como son la aplicación de fuerzas, motores o relaciones de posición han sido los mismos y en base a estos, se ha buscado cuales dependen de la aeronave y cuáles no.

1.4. Objetivos del trabajo

El objetivo de este proyecto ha sido parametrizar el proceso de simulación. Desde que se ejecuta la simulación de fluidos con el complemento de SOLIDWORKS® Flow Simulation, hasta que se ejecuta la simulación cinemática y dinámica con el complemento Motion, pasando por los tratamientos de datos intermedios y finales que es necesario hacer.

De esta manera un usuario que quiera conocer qué aeronave le va a proporcionar mejores resultados, simplemente tendrá que introducir los parámetros requeridos, calcular y decidir cuál será la mejor para su experimento y qué parábola trazar.

O desde otro punto de vista, según la aeronave que el usuario disponga, se pueda saber qué trazada será la óptima.

2.La microgravedad y el vuelo parabólico

2.1. Principios básicos del vuelo

Es necesario, en primer lugar, conocer los ejes imaginarios y movimientos de una aeronave [3].

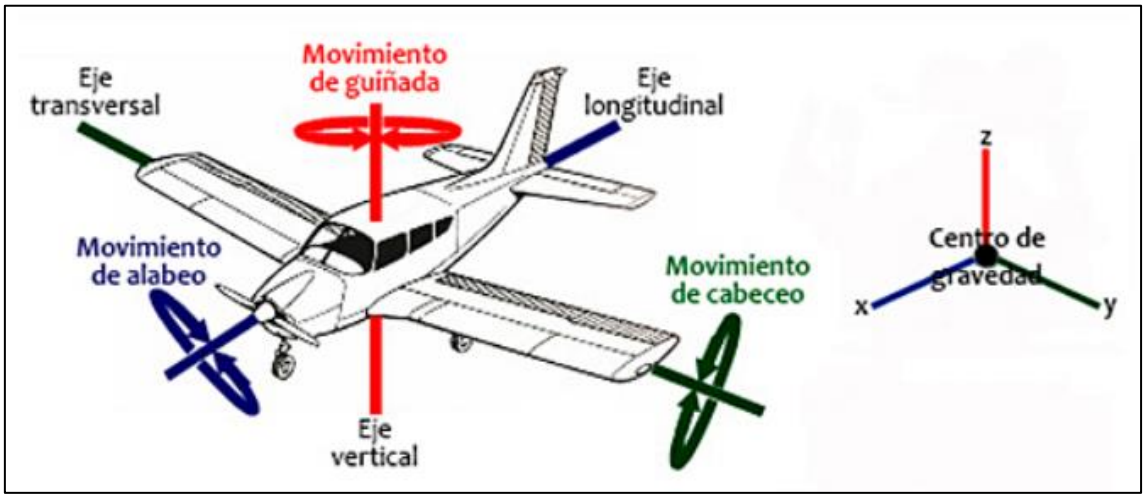


Figura 1. Ejes y movimientos de una aeronave.

Eje longitudinal: va desde el morro hasta la cola del avión. El movimiento alrededor de este eje se denomina alabeo.

Eje transversal o lateral: va desde el extremo de un ala hasta el extremo opuesto de la otra. El movimiento alrededor de este eje se denomina cabeceo.

Eje vertical: atraviesa el avión por su centro de gravedad formando 90º con los otros dos ejes. El movimiento alrededor de este eje se denomina guiñada.

También es necesario, antes de comenzar, introducir varios conceptos para la buena comprensión del simulador [2].

Actitud de la aeronave: orientación o referencia angular de los ejes longitudinal y trasversal de la aeronave con respecto del horizonte, que se especifica en los términos de posición del morro y posición de las alas.

Trayectoria de vuelo: es el recorrido que lleva a cabo la aeronave. Es importante no asociar la trayectoria con la actitud, por ejemplo, en una trayectoria de vuelo recto y nivelada la aeronave posiblemente tenga una actitud de morro ligeramente elevada. En este caso, se le llamará también trazado, para hacer referencia a las parábolas.

Viento relativo: flujo de aire que produce la aeronave cuando se desplaza. Es paralela a la trayectoria y de dirección contraria. Su velocidad es la que tiene la aeronave con relación a la masa de aire en la que se mueve, no respecto de la tierra.

Ángulo de ataque: es el ángulo formado entre la cuerda del ala y la trayectoria de la aeronave. Este ángulo puede ser modificado por el piloto. Al modificarlo se varía la inclinación de la aeronave con respecto al viento relativo y resulta uno de los parámetros más importantes a la hora de volar.

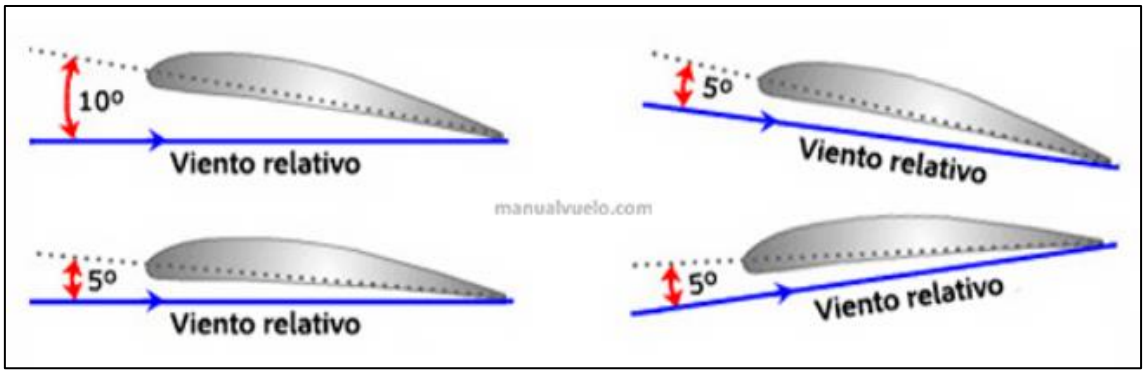


Figura 2. Ángulo de ataque según el viento relativo.

Las fuerzas principales que actuarán sobre la aeronave, que será útil conocer para establecer los parámetros cinemáticos del módulo de Motion, son las siguientes [3]:

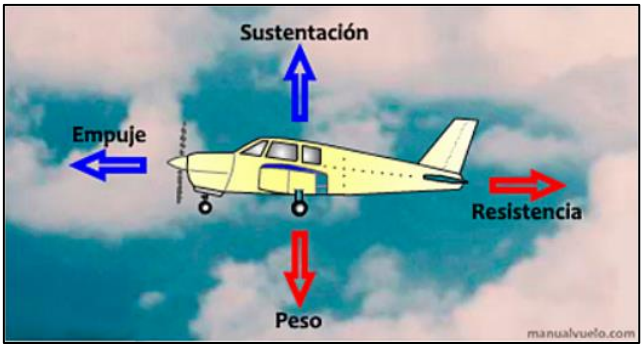


Figura 3. Fuerzas principales que actúan sobre un avión.

Sustentación: es la fuerza desarrollada por un perfil aerodinámico moviéndose en el aire, ejercida de abajo arriba y cuya dirección es perpendicular al viento relativo y a la envergadura de la aeronave (no necesariamente perpendiculares al horizonte). Se suele representar con la letra L.

Peso: es la fuerza que ejerce la atracción gravitatoria sobre la aeronave. Su dirección es perpendicular a la superficie de la tierra y ha de ser compensada por la sustentación para mantener el avión en el aire.

Resistencia: es la fuerza que actúa de forma paralela y en la misma dirección que el viento relativo, pero en sentido opuesto. Se puede decir que impide o retarda el movimiento de la aeronave

Empuje: también llamado impulso durante este trabajo. Fuerza necesaria para vencer a la resistencia. Depende, principalmente, de la potencia del motor de la aeronave.

2.2. La microgravedad

El término microgravedad hace referencia a un entorno en el que la cantidad de fuerzas g existentes se aproximan a un valor nulo, sin llegar a serlo, como el que podrían llegar a tener los astronautas en el espacio [4]. Se define a 1 fuerza g como una magnitud que equivale a la aceleración de la gravedad en la superficie de la Tierra, que es 9,81 m/s2.

En la realidad resulta imposible experimentar la gravedad nula, ya que habría que estar lo suficientemente lejos de cualquier cuerpo del universo para que ninguno afectara. Si se escapa lo suficientemente lejos de la Tierra, estaría el Sol, que también ejerce su gravedad, y si es más lejos, estaría la Vía Láctea. Pero el Universo está lleno de Galaxias como la Vía Láctea, por tanto, sería inviable.

No es posible alcanzar el valor nulo, pero sí aproximarse a la gravedad existente en planetas cercanos, que es lo que interesa. Tener un entorno donde la gravedad sea más baja que en la superficie terrestre puede ser útil para conocer como reaccionan los seres vivos y las infraestructuras por ellos creadas para una futura colonización de otros planetas.

2.3. Métodos de obtención

Estación espacial

Es el entorno en el que el valor de la gravedad se puede aproximar más a cero. La ISS (Estación Espacial Internacional) es un centro de investigación, situado en una orbita a 408 km de la superficie terrestre, que garantiza la presencia continua del ser humano en el espacio.

Allí se llevan a cabo experimentos de diversa consideración, que suponen gastos más elevados para las entidades que los realizan.

Torre de caída libre

Es un edificio, concretamente una torre, situada en la superficie terrestre. Desde su punto más alto se lanzan los elementos a estudiar y se consiguen fuerzas g del orden de 10^{-3} a 10^{-5}.

En la torre de caída libre de Bremen (Alemania) (Figura 4), que posee 122 m de caída hábiles, se puede conseguir la que el experimento sufra microgravedad durante 4,74 segundos [5].

Figura 4. Torre de caída libre de Bremen.

Vuelos parabólicos

Mediante la ejecución de maniobras acrobáticas, en este caso parábolas, con aviones civiles o con avionetas deportivas se procura alcanzar las condiciones de microgravedad.

Se trata de un vuelo en el que, una vez se estabiliza la aeronave a una altura segura, se produce la ascensión con una inclinación de 45º – 47º, momento en el que comienza un periodo de sobregravedad, alcanzando valores de 1,5 g a 2 g. A continuación, se reduce la potencia y se corrige la inclinación para trazar la parábola, periodo en el que se experimenta la denominada microgravedad alcanzando valores fuerza g del orden de 10^{-2} g. Finalmente, se produce el descenso con la misma inclinación de ascenso, pero negativa, volviendo a experimentar la sobregravedad antes de estabilizar otra vez la aeronave.

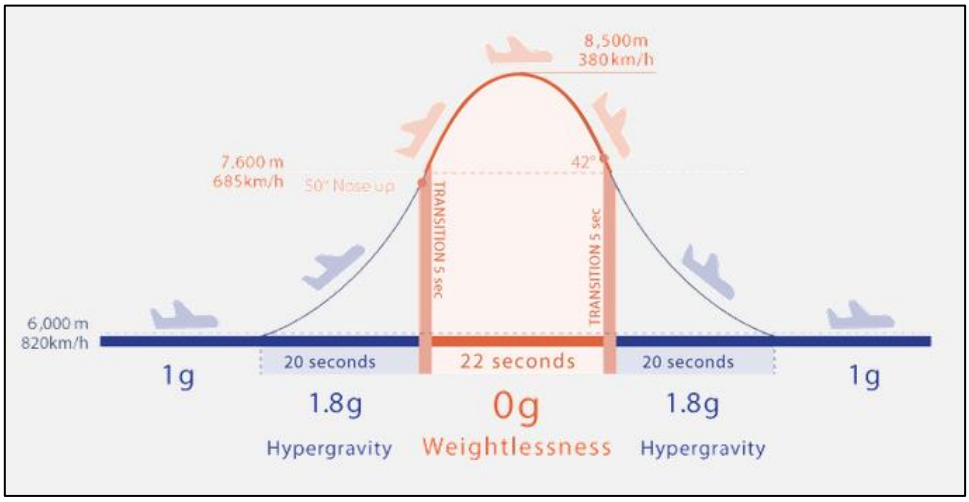


Figura 5. Vuelo parabólico con un A310.

Dependiendo de la aeronave con la que se trabaje, los resultados serán diferentes. Un avión del tipo A310, como los que usan entidades como AirZero-G [6], puede alcanzar los 22 segundos de ingravidez, pero una avioneta acrobática por su evidente menor potencia, reduciría el tiempo a la mitad.

Aparte de vuelos experimentales, también se llevan a cabo este tipo de viajes como actividad de ocio.

3. Simulación con SOLIDWORKS®

El software SOLIDWORKS® es una herramienta de diseño asistido por ordenador (CAD) paramétrica. El modelado paramétrico permite la reutilización de elementos ya existentes y la rápida modificación del diseño basada en los resultados del análisis de ingeniería [7].

Esta herramienta permite el modelado de piezas y ensamblajes en 3D, así como la obtención de sus planos técnicos en 2D. Pero además, existen diversas extensiones (Figura 6), aquí llamadas complementos (o *Add-in* en inglés); por ejemplo: SOLIDWORKS CAM, para evaluar las posibilidades de fabricación; SOLIDWORKS *Electrical*, para integrar esquemas eléctricos en el producto o SOLIDWORKS *Simulation*, que utiliza el análisis de elementos finitos (FEA) para predecir el comportamiento físico del elemento, entre otras.

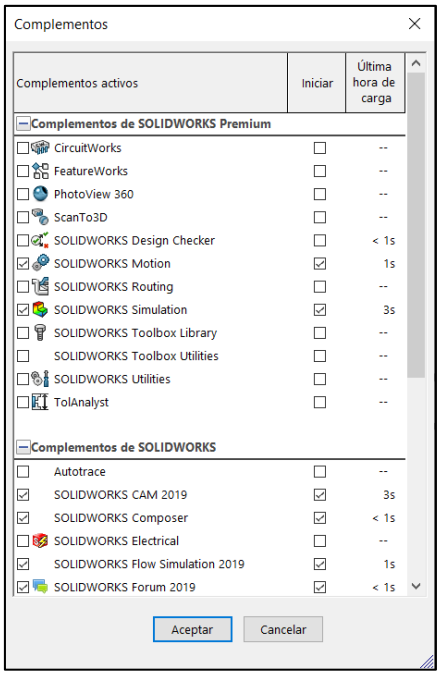


Figura 6. Selección de complementos de SOLIDWORKS.

En el caso de esta línea de investigación, la simulación llevada a cabo necesitará, a parte del modelado en 3D de la aeronave, los complementos *Flow Simulation* y *Motion*. A continuación, explica qué herramientas proporcionadas en cada complemento serán necesarias para completar la simulación.

3.1. Simulación con Flow Simulation

SOLIDWORKS® *Flow Simulation* es una solución de dinámica de fluidos computacional (CFD) que permite simular de forma rápida y sencilla flujos de líquido y gas a través y alrededor de los diseños para poder calcular así el rendimiento y las capacidades del producto [8].

El proyecto comienza con un estudio tipo túnel de viento, en el que el sólido, en este caso la aeronave, es sometida a condiciones aerodinámicas que experimentará en la realidad.

Mientras la aeronave permanece estática, con el eje longitudinal en posición horizontal, se propulsa aire a diferentes ángulos con respecto a la horizontal y para cada ángulo, el aire es propulsado a diferentes velocidades. Es decir, si se toma como ejemplo el trabajo del Sukhoi-26,

se hizo un análisis para los ocho ángulos de ataque de valor 0º, 3º, 4º, 6º, 8º, 10º, 12º y 14º. De cada ángulo se hicieron seis estudios, cada uno a una velocidad de 200, 250, 300, 350, 400 y 450 km/h. Obteniendo, de esta manera, un total de 48 estudios individuales.

Los parámetros para introducir en el proyecto son:

CONDICIONES GENERALES

Pueden ser introducidas a través de un asistente (o *wizard* en inglés), bastante completo, que guiará al usuario. (Figura 7)

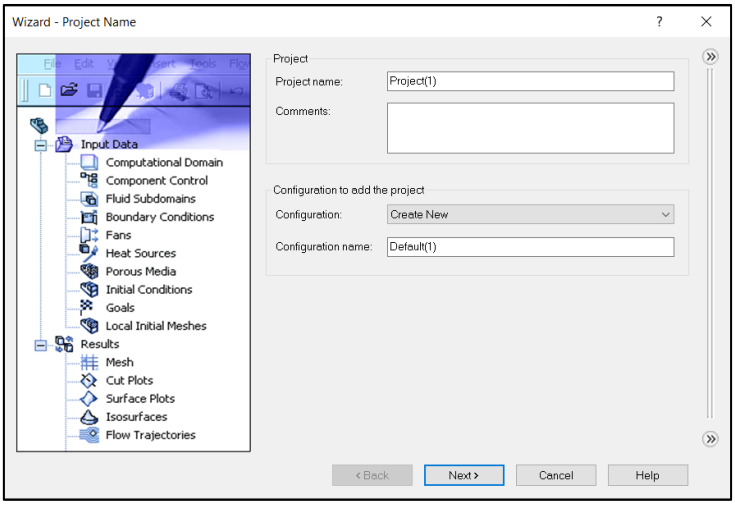


Figura 7. Asistente de configuración del estudio de fluido.

Tipo de análisis		
Tipo de análisis		**Externo**
Excluir cavidades sin condiciones de flujo		☑
Excluir espacios internos		☑
Fluidos		
Fluido		**Aire (gases)**
Tipo de fluido		**Laminar y turbulento**
Condiciones de la pared		
Condición térmica de la pared		**Pared adiabática**
Rugosidad		**0 micrómetros**
Condiciones iniciales y de ambiente		
Parámetros termodinámicos	Parámetros	**Presión, temperatura**
	Presión	**79540.125 Pa.**
	Temperatura	**280.2 K.**
Parámetros de velocidad	Parámetro	**Velocidad**
	Definido por	**Ángulos aerodinámicos**
	Velocidad	**-200 km/h.**
	Plano longitudinal	**YZ**
	Eje longitudinal	**Z**
	Ángulo de ataque	**0 rad.**
	Ángulo de desplazamiento lateral	**0 rad.**

Tabla 1. Condiciones generales

CONDICIONES ESPECÍFICAS

Se han de introducir manualmente desde las pestañas *Computational Domain*, *Calculation Control Options, Global Mesh Settings* y *Goals*.

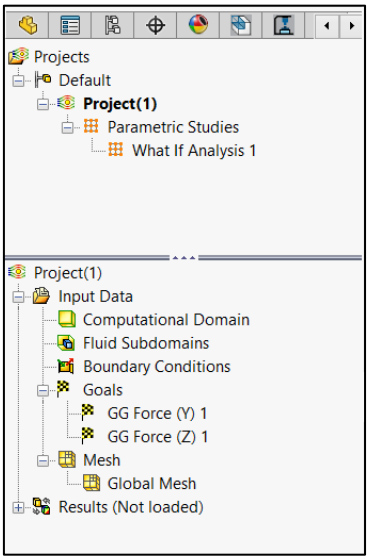


Figura 8. Parámetros de proyecto de Flow Simulation.

Dominio computacional		
Plano X = 0 Positivo		**Variable. Dependiente de las dimensiones de la aeronave**
Plano X = 0 Negativo		**Variable. Dependiente de las dimensiones de la aeronave**
Plano Y = 0 Positivo		**Variable. Dependiente de las dimensiones de la aeronave**
Plano Y = 0 Negativo		**Variable. Dependiente de las dimensiones de la aeronave**
Plano Z = 0 Positivo		**Variable. Dependiente de las dimensiones de la aeronave**
Plano Z = 0 Negativo		**Variable. Dependiente de las dimensiones de la aeronave**
Opciones de control del cálculo		
Condiciones de parada de cálculo	Convergencia de los objetivos	☑
	Travels	☑ **[automático]**
Objetivos		
Parámetros		**Fuerza Y** ☑
		Fuerza Z ☑
Ajustes de malla global		
Tipo		**Automático**
Nivel de la malla inicial		**6**
Ratio		**1**
Refinamiento avanzado		☑

Tabla 2. Condiciones específicas.

WHAT IF ANALYSIS

Además, SOLIDWORKS® nos proporciona una herramienta que no se menciona en el trabajo tomado como referencia [2], pero que puede ser de gran utilidad. Se trata del "Análisis Y Si" (o *What If Analysis* en inglés).

Con ella, indicando las variables de entrada y los objetivos de salida, se computan todos los estudios de manera continuada. Así se evitará tener que estar pendiente del software cada vez que acaba una simulación.

		Ángulo de ataque (grados)				
Velocidad (km/h)		Depende de las capacidades de la aeronave				
		Variable 1	Variable 2	Variable 3	Variable 4	Variable…
Depende de las capacidades de la aeronave	Variable 1	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z
	Variable 2	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z
	Variable 3	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z
	Variable 4	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z
	Variable …	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z	Fuerza Y – Fuerza Z

Tabla 3. Estudio paramétrico de SOLIDWORKS.

3.2. Tratamiento intermedio de datos

Las fuerzas obtenidas son las que experimenta la aeronave a lo largo del eje longitudinal y a lo largo del eje vertical. Pero no son estas las necesarias para continuar con el estudio de movimiento en el complemento *Motion*, sino la fuerza de sustentación y la de resistencia.

Figura 9. Fuerzas que actúan sobre la aeronave.

Estas dos fuerzas dependen del ángulo de ataque que tenga la aeronave o lo que es lo mismo, del grado de cabeceo al que esté sometida. Su cálculo es posible con las siguientes ecuaciones:

Sustentación

$$L = \cos(\alpha) \cdot Fy - \sin(\alpha) \cdot Fz$$

Resistencia

$$Fr = \sin(\alpha) \cdot Fy - \cos(\alpha) \cdot Fz$$

donde α es el ángulo de ataque.

Una vez se hayan obtenido todas las fuerzas de sustentación y resistencia, podemos obtener sus respectivos coeficientes. Teniendo en cuenta el modelo matemático de estas dos, si agrupamos términos y despejamos, queda lo siguiente:

Coeficiente de sustentación

$$L = \frac{\rho \cdot v^2 \cdot S \cdot Cl}{2} \implies L = Kl \cdot v^2 \implies Kl = \frac{L}{v^2}$$

donde ρ es la densidad del aire; v la velocidad relativa de la aeronave; S, la superficie alar y Cl, el coeficiente de sustentación (dependiente del ángulo de ataque).

Coeficiente de resistencia

$$D = \frac{\rho \cdot v^2 \cdot S \cdot Cd}{2} \implies L = Kd \cdot v^2 \implies Kd = \frac{D}{v^2}$$

donde ρ es la densidad del aire; v la velocidad relativa de la aeronave; S, la superficie alar y Cl, el coeficiente de sustentación (dependiente del ángulo de ataque).

Calculando otra vez todos los valores, para cada ángulo de ataque se tendrá tantos coeficientes de sustentación y de resistencia como velocidades se hayan estudiado. El siguiente paso es hacer la media aritmética de los coeficientes para cada ángulo.

De esta manera, si se grafican los coeficientes de resistencia en función del ángulo por una parte y los coeficientes de sustentación en función del ángulo por otra, se puede trazar la línea de tendencia y extraer la ecuación que la representa (Figura 10).

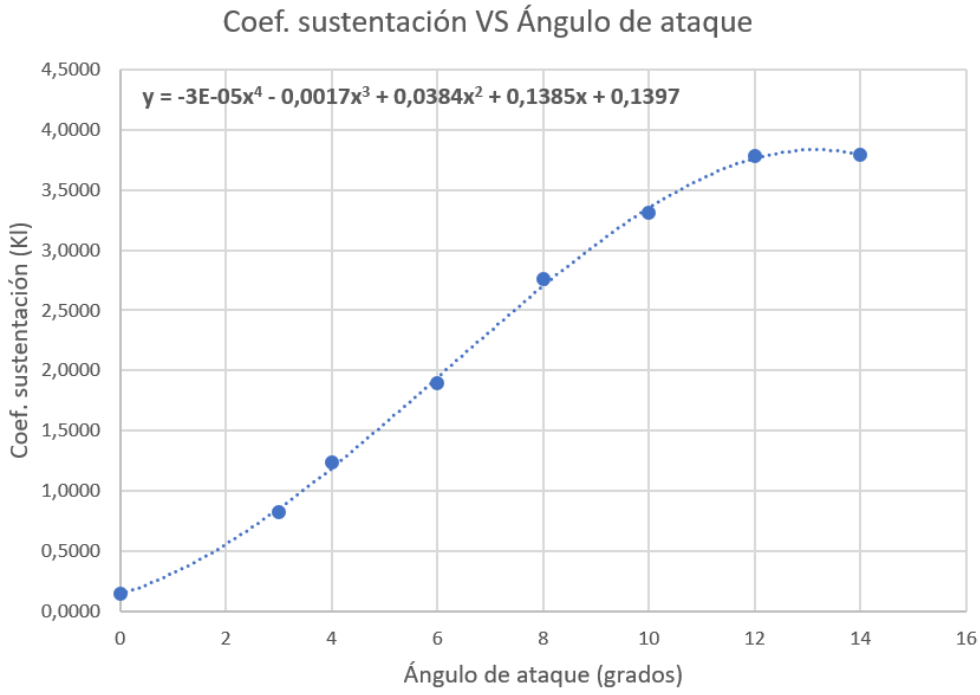


Figura 10. Ejemplo de gráfica para la extracción de la ecuación polinómica

Estas dos ecuaciones, que informan del coeficiente según el ángulo de ataque, serán necesarias para trabajar en el complemento *Motion*. Serán dos de las variables en esa parte del proyecto.

3.3. Simulación con Motion

SOLIDWORKS® *Motion* es posible llevar a cabo simulaciones gráficas de movimiento para modelos de ensamblaje basándose en las ecuaciones de la cinemática y dinámica de cuerpos rígidos.

Se utilizará esta herramienta con el modelo simplificado de aeronave creado por el profesor Miguel Brigos.

Este modelo consta de tres piezas (Figura 11): cabina, contenido y cinta de velocidad; que unidas en un ensamblaje formarán un conjunto al que se le aplicarán los efectos de movimiento disponibles como fuerzas, motores o la propia gravedad. Así, se procurará una aproximación a un vuelo real.

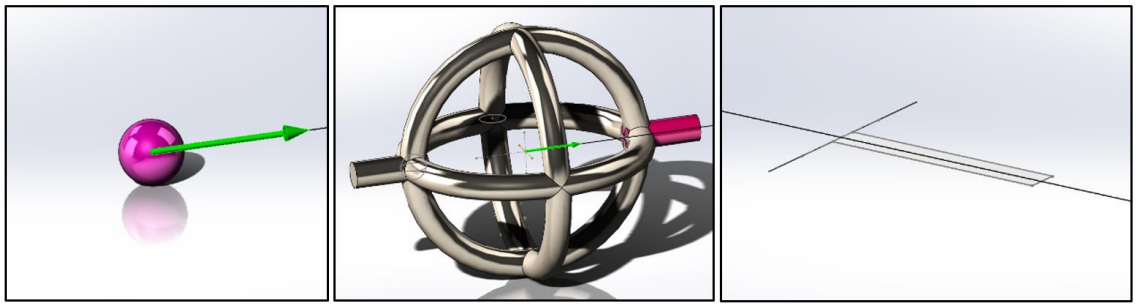


Figura 11. Contenido, cabina y cinta de velocidad.

Los parámetros para introducir en el proyecto son:

Parámetro	Tipo	Definido por
Aceleración de gravedad	Gravedad	Constante
Sustentación	Fuerza	Expresión. Dependiente de la aeronave
Resistencia total	Fuerza	Expresión. Dependiente de la aeronave
Impulso	Fuerza	Segmentos. Dependiente de la aeronave
Velocidad inicial de vuelo	Motor lineal	Constante. Dependiente de la aeronave
Dirección del viento	Motor rotatorio	Constante
Dirección del viento	Motor rotatorio	Expresión
Ángulo de ataque	Motor rotatorio	Segmentos. Dependiente de la aeronave

Tabla 4. Parámetros del estudio en Motion.

3.4. Enfoque global del proyecto

Por tanto, se puede decir que el estudio del vuelo parabólico de una aeronave está formado por cuatro partes.

En la primera se analiza el comportamiento del modelo CAD a diferentes ángulos de ataque y velocidades. (En color azul en el esquema de la Figura 12)

En la segunda se tratan los datos provenientes del estudio del fluido, para posteriormente insertarlos en el estudio de movimiento. (En color verde en el esquema de la Figura 12)

En la tercera se analiza el comportamiento cinemático y dinámico de un modelo simplificado de la aeronave. (En color rojo en el esquema de la Figura 12)

En la cuarta se vuelven a tratar los datos, en este caso los del estudio de movimiento, para conocer la altura alcanzada por la aeronave y la fuerza g en cada instante. (En color verde en el esquema de la Figura 12)

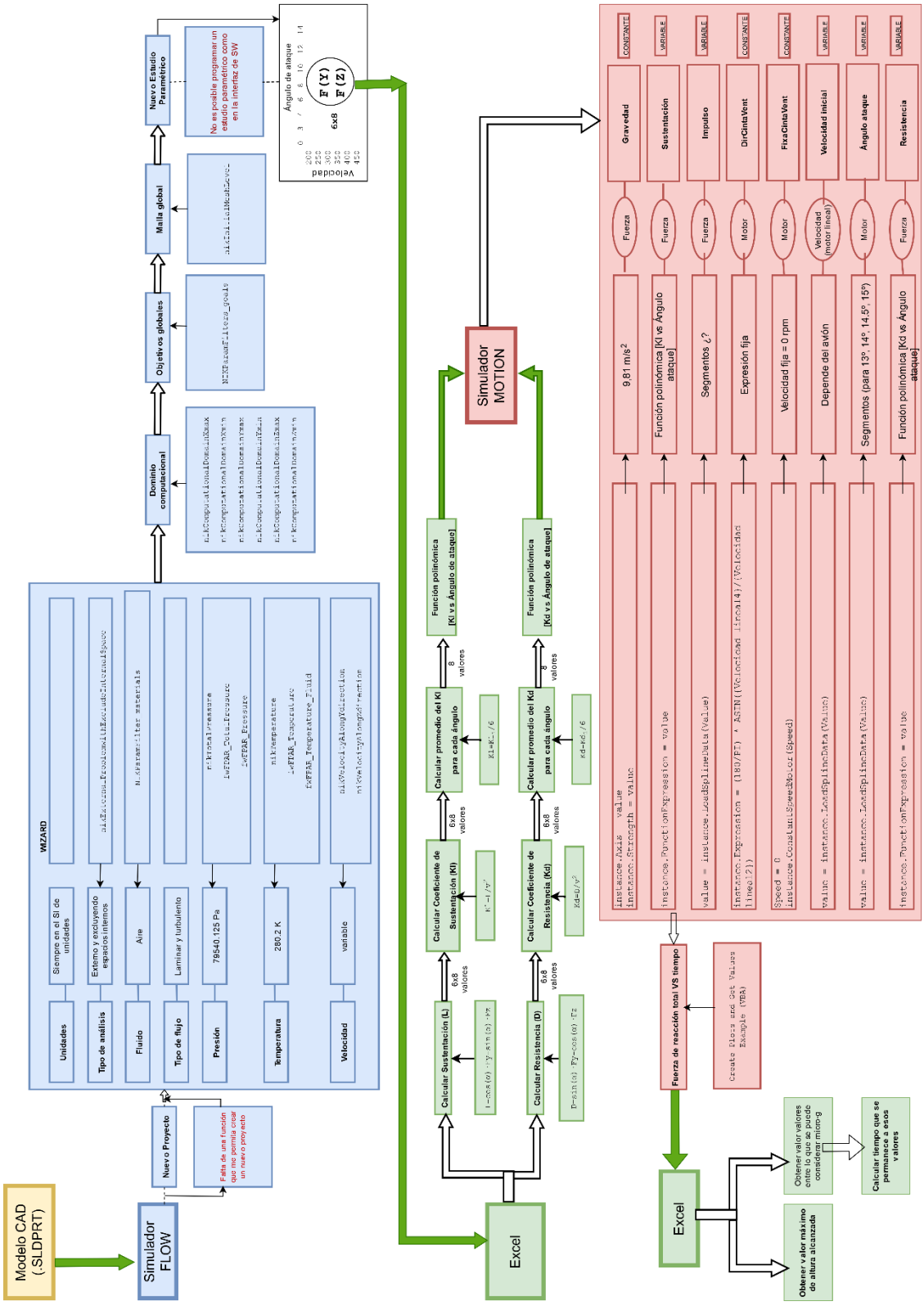


Figura 12. Esquema del estudio de un vuelo parabólico.

4.El programa

4.1. La API (Application Programming Interface)

Una interfaz de programación de aplicaciones (o API, por sus siglas en inglés), es un conjunto de subrutinas, funciones y métodos que ofrece una biblioteca para ser utilizada por otro software como capa de abstracción. [9]

De esta forma, un programador llama a una biblioteca para utiliza los servicios que esta le ofrece sin tener que programar todos los servicios desde cero.

La API de SOLIDWORKS® contiene una gran cantidad de funciones que pueden ser llamadas desde los lenguajes de programación Visual Basic para Aplicaciones (VBA), VB.NET, Visual C#, Visual C++ 6.0, y Visual C++/CLI. Estas funciones permiten acceder a las funcionalidades y propiedades de SOLIDWORKS como pueden ser dibujar un plano en 2D, extruir una pieza en 3D o editar ciertos parámetros del complemento de análisis de elementos finitos.

Todos estos métodos, propiedades y constantes que SOLIDWORKS® utiliza, están disponibles en su web [10] completamente actualizados o en los archivos instalados junto con la aplicación. Además, hay algunos ejemplos para utilizar como apoyo.

Sin embargo, la documentación bibliográfica y la información disponible en internet no es extensa, y aproximadamente el 80 % de ella se centra en la automatización de la parte de modelado o trazado de planos. A diferencia de la aplicación de Microsoft, Excel, sobre la que se puede encontrar todo tipo de explicaciones y subrutinas presumiblemente, porque la cantidad de usuarios es mucho mayor.

Este proyecto está basado en la programación sobre los complementos *Flow Simulation* y *Motion*. Hecho que ha complicado la programación ya que, probablemente la posición que adopta un ingeniero a la hora de confiar en un simulador hace que no sean tantos los ejemplos y la documentación sobre el tema. Es decir, se prefiere reservar los aspectos más delicados de la ingeniería a la intervención humana.

Esta coyuntura ha hecho que, al llegar a cierto punto de estancamiento, se decidiera descartar la parte de *Flow Simulation*. Por la falta de algunas funcionalidades sumada a la documentación exclusiva en dos lenguajes de programación diferentes al elegido para hacer la aplicación, se tomó la decisión.

4.2. Tipos de aplicaciones

Para desarrollar una aplicación es importante plantearse de qué tipo va a ser. En este caso se nos presentan tres opciones [11]: un macro, un complemento (o *add-in*, en inglés) o un programa independiente (o *stand-alone*, en inglés).

- Macro de SOLIDWORKS®
  Es la forma más rápida y sencilla de comenzar a programar en SOLIDWORKS®, motivo por el cual se comenzó a hacer este proyecto en ese lenguaje.
  Microsoft VBA es una serie de herramientas basadas en Microsoft Visual Basic para Aplicaciones (VBA), que están incluidas en el propio software de SOLIDWORKS®. Con

ellas es posible grabar, ejecutar y editar macros exclusivamente con el lenguaje VBA en el propio SOLIDWORKS®. Se guardarán en formato **.swp**.

- Aplicación Stand-alone
  Se trata de una aplicación (**.exe**) que no requiere ser instalada y que puede funcionar de forma totalmente independiente con respecto a SOLIDWORKS®. Es decir, que no es una extensión del software, sino que se conecta a él cuando es requerido.
  Permite utilizar cualquier lenguaje de programación que soporte la plataforma de Microsoft COM (Component Object Model, en inglés):
  - Visual Basic .NET (VB.NET)
  - Visual C++/CLI
  - Visual C#.NET
  - Visual C++ 6.0
- Add-in
  Es un complemento integrado disponible para el usuario, que se añade o no al iniciar SOLIDWORKS®. *Flow Simulation* o *Motion*, serían ejemplos de aplicaciones del tipo add-in. Se crea en la extensión **.dll**, pero es necesario un paquete instalador para que el usuario lo pueda obtener.
  Permite utilizar cualquier lenguaje de programación que soporte la plataforma de Microsoft COM (Component Object Model, en inglés):
  - Visual Basic .NET (VB.NET)
  - Visual C++/CLI
  - Visual C#.NET
  - Visual C++ 6.0

Elección de la estructura

En primer lugar, cabe destacar la decisión del lenguaje para programar. En un primer momento se decidió usar VBA por su sencillez y accesibilidad, pero la complejidad del proyecto hizo que se cambiara a Visual Basic .NET.

El lenguaje VBA puede resultar muy útil en las aplicaciones de Microsoft Office, pero a pesar de que también esté integrado en SOLIDWORKS®, tiene aspectos negativos que obligan a desechar esta opción. Entre otras: un pobre soporte para la programación orientada a objetos [12], entradas de registro COM complejas y frágiles [13] o la escasa documentación en relación con SOLIDWORKS®.

A la hora de programar el módulo de *Motion* surgieron varios problemas de compilación, debidos en gran parte a fallos conceptuales, pero también a errores sintácticos. Este hecho supuso tener que rebuscar información y utilizar una de las herramientas que SOLIDWORKS® nos proporciona, que no es otra que una plantilla para crear un add-in (complemento). [14] [Disponible en **SOLIDWORKS Downloads\SOLIDWORKS 20XX\apisdk\SolidWorksAPI SDK.msi**]

De esta manera, se decidió ir por la línea de una aplicación del tipo *add-in*, habiendo tenido en cuenta, además, sus aspectos positivos como:

- Puede usar todas las API de SOLIDWORKS®.
- A pesar de ser más difícil de comprender, al comenzar desde una plantilla se evita tener problemas por falta de declaraciones o simples descuidos.
- Está integrada de manera cómoda para el usuario dentro de SOLIDWORKS®.

- Puede usar *call-backs* (devoluciones de llamada)
- Aunque, a diferencia de la aplicación *stand-alone*, esta requiera elaborar un paquete instalador, no se trata de un proceso complejo.

El desarrollo del código se completó en el IDE (Entorno de desarrollo integrado) gratuito Visual Studio 2019.

En líneas generales, teniendo en cuenta las decisiones y descartes, la idea del programa estaba basada en un acceso a Excel para generar un informe y hacer los cálculos previos al estudio de movimiento en primer lugar (en verde en la Figura 12). Después, pedir al usuario los parámetros variables (o datos) necesarios a través de un formulario. Proceder a entrar al complemento Motion para editar todos los parámetros necesarios (en rojo en la Figura 12); y finalmente, trasladar los resultados de la simulación al informe creado al inicio (en verde en la Figura 12).

4.3. Diagrama del modelo de objetos

El diagrama de la Figura 13 muestra la relación entre las interfaces del modelo de objetos de la API de SOLIDWORKS® [15]. Están solo presentes las interfaces utilizadas, con sus respectivas propiedades y funciones.

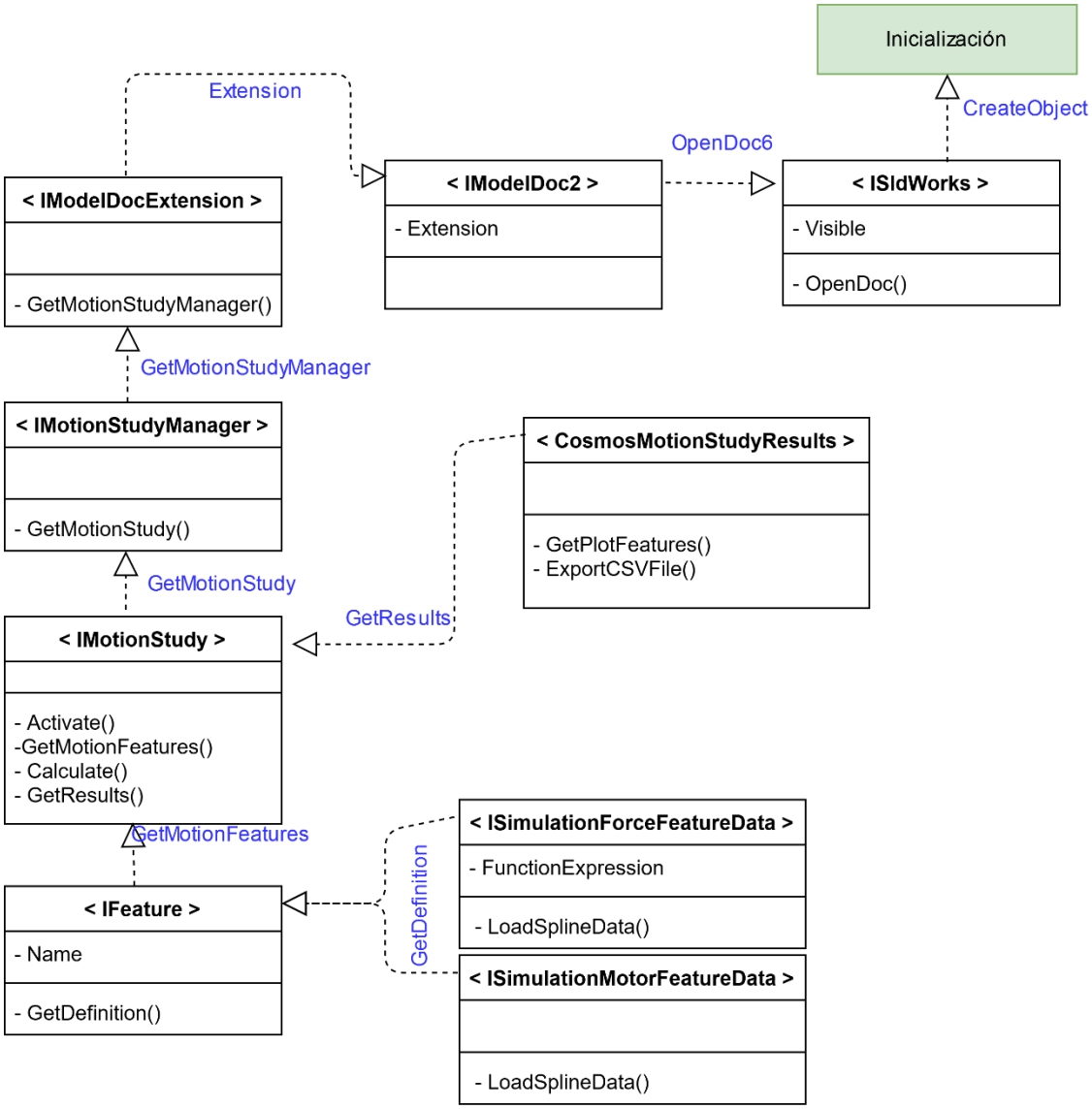


Figura 13. Diagrama del modelo de objetos de SOLIDWORKS.

Leyenda:

Relación entre interfaces a través de propiedades o métodos

< ISldWorks >

Nombre de la interfaz

- Visible

Propiedades utilizadas en la interfaz

- OpenDoc()

Métodos utilizados en la interfaz

4.4. Formulario

El formulario será la interfaz de comunicación del usuario con la aplicación creada. A su través, se introducirán los datos y se pedirá al programa que comience la simulación.

4.4.1. Diseño

Primero, se pensó un posible formulario y se hizo un boceto sencillo sobre papel.

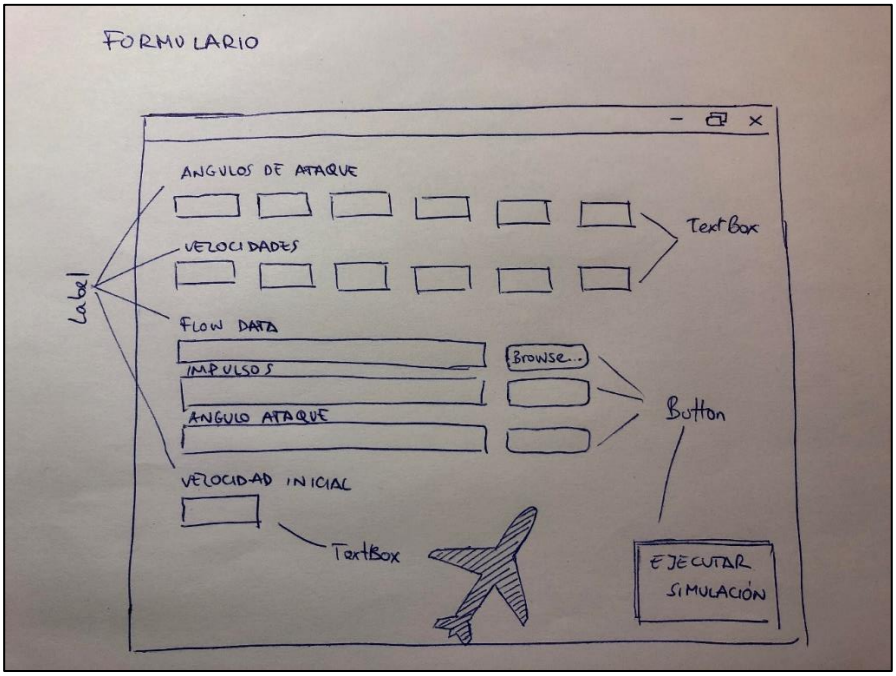


Figura 14. Boceto de formulario.

Finalmente, el proyecto acaba con un formulario semejante al inicial, añadiendo un parámetro más de entrada que no se había tenido en cuenta al principio.

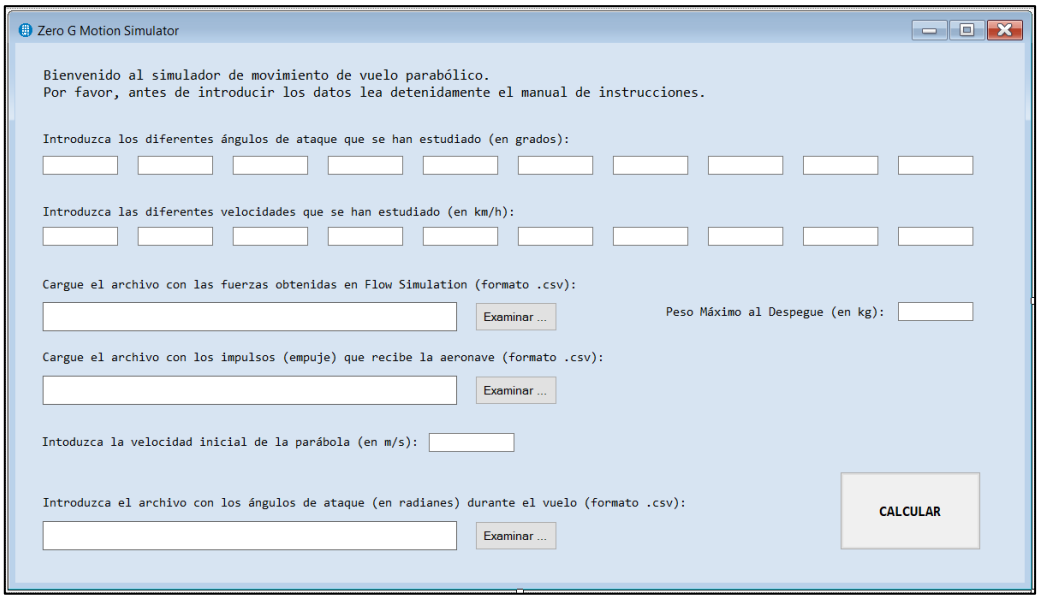


Figura 15. Formulario final.

4.4.2. Datos de entrada

Los datos de entrada (*input*) han sido seleccionados por ser los parámetros dependientes de la aeronave escogida. Es decir, son aquellos que el usuario tendrá que aportar, que dependerán de las capacidades de maniobra de la aeronave, de su potencia o de su estructura. En los apartados 3.1 y 3.3 (Tablas 2 y 4) se han coloreado de amarillo para diferenciarlos de los que permanecen constantes en todos los proyectos.

Estos se pueden dividir en dos partes, según se piden en el formulario (Figura 15):

Los que utiliza Excel

- 10 TextBox para introducir los ángulos de ataque estudiados en Flow Simulation.
- 10 TextBox para introducir todas las velocidades estudiadas para cada ángulo de ataque.
- 1 Button para cargar el archivo (.csv) con las fuerzas obtenidas en Flow Simulation.

Los que utiliza SOLIDWORKS®

- 1 Button para cargar el archivo (.csv) con los impulsos que recibe la aeronave en el vuelo.
- 1 Button para cargar el archivo (.csv) con los ángulos de ataque de la aeronave en cada momento del vuelo.
- 1 TextBox para introducir la velocidad inicial al comenzar la maniobra.
- 1 TextBox para introducir la masa de la aeronave

4.5. Código

Todo lo aquí citado hace referencia al código, presente en ANEXO I. El propio código posee también los comentarios procedentes que se han ido haciendo durante el desarrollo.

El código de esta aplicación está formado por dos funciones importantes: ZeroGMotionSimulation() y Calculo(). Esta última, que contiene el grueso del desarrollo, formada, a su vez, por tres partes diferenciadas.

4.5.1. ZeroGMotionSimulation()

[Líneas 522 – 528]

Esta función conecta el *add-in* con el formulario (Figura 15). Así, al ejecutar el único botón presente en la barra de herramientas del *add-in* (Figura 16), aparecerá automáticamente el formulario para que el usuario pueda introducir los datos.

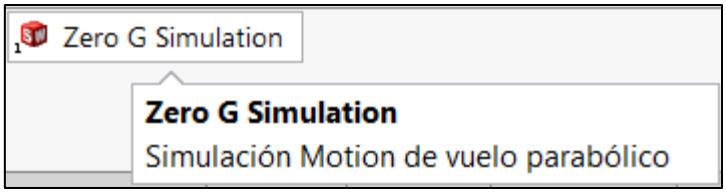


Figura 16. Botón del add-in.

4.5.2. Calculo()

[Líneas 2 – 520]

4.5.2.1. Cálculos previos

[Líneas 3 – 238]

En primer lugar, se toma el documento Excel, denominado "Infome.xlsx", que está contenido en la carpeta "Archivos" de la aplicación. A él, se le añaden dos hojas de cálculo que servirán para depositar posteriormente más datos.

Se toman los ángulos de ataque y velocidades que ha introducido el usuario a través del formulario, así como el archivo de datos con las "Fuerzas Y" y "Fuerzas Z", procedentes del estudio con *Flow Simulation* . Con estos datos se hacen todos los cálculos mencionados en el apartado 3.2, con el fin de obtener la función polinómica para editar los parámetros Sustentación y Resistencia del tipo fuerza (Tabla 3).

4.5.2.2. SOLIDWORKS® API

[Líneas 240 – 411]

Se carga el ensamblaje de SOLIDWORKS ®, denominado "SimuladorVueloParam.SLDASM", que se encuentra en la carpeta "Archivos" de la aplicación.

Se accede a la interfaz MotionStudy, que representa el estudio de movimiento ya creado dentro del ensamblaje y a su vez, dentro de esta, se accede a la interfaz Feature, donde están los parámetros que se van a editar. Tres parámetros del tipo fuerza y dos del tipo motor, uno rotatorio y otro lineal (de amarillo en la Tabla4).

A la hora de editar los parámetros "Velocidad inicial de vuelo", "Impulso" y "Ángulo de ataque" resulta necesario hacer una adaptación, ya que no se encuentra la forma en que las interfaces SimulationMotorFeatureData y SimulationForceFeatureData permitan tanto la definición por segmentos como la edición de la velocidad constante. Se decide finalmente introducirlos a través de archivos de datos (LoadSplineData()).

Las funciones polinómicas de los parámetros "Resistencia" y "Sustentación" sí que permiten ser introducidos a través de la propiedad habitual FunctionExpression.

Parámetro	Definido por	Definida en la aplicación por
Sustentación	Expresión	Espresión
Resistencia	Expresión	Expresión
Impulso	Segmentos	Archivo de datos (.csv)
Velocidad Inicial	Valor constante	Archivo de datos (.csv)
Ángulo de ataque	Segmentos	Archivo de datos (.csv)

Tabla 5. Parámetros variables para programar.

El paso final en SOLIDWORKS® es ejecutar la simulación y extraer los resultados en la extensión .csv. En este caso, se extraen la atura de la aeronave y la fuerza total de reacción en función del tiempo.

4.5.2.3. Cálculos finales

[Líneas 412 – 519]

Se vuelve a activar el documento Excel y se colocan los datos de la altura en la hoja 3 y los datos de la fuerza total de reacción en la hoja 2. Para estos últimos, por su utilidad, se calcula la reacción en unidades de fuerza g, utilizando el dato del peso de la aeronave.

Finalmente, se crean dos gráficos en la hoja nueva para visibilizar los nuevos resultados

4.6. Instalador

Una vez se ha finalizado el desarrollo de la solución y hay que entregárselo al usuario, resulta imprescindible crear un instalador para hacer la experiencia lo más "agradable" posible.

Por tanto, con la ayuda de la extensión Visual Studio Installer (VSI), proporcionada por Visual Studio 2019, se crea un paquete instalador [16] (extensión .msi) para que la difusión de la aplicación sea más sencilla y siguiendo las instrucciones del Manual de usuario (Volumen III), se pueda obtener en cualquier ordenador personal.

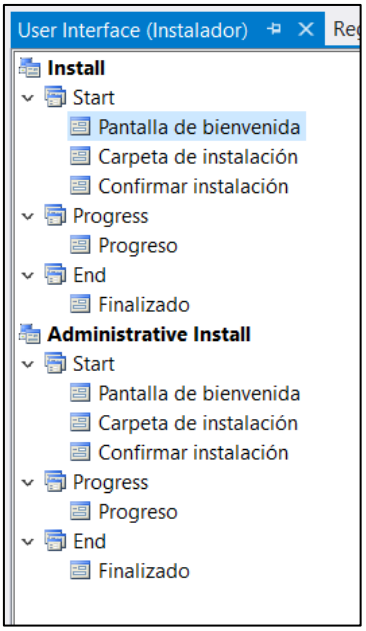


Figura 17. Interfaces del instalador.

4.7. Recopilación de problemas y soluciones a lo largo del desarrollo

Problema 1:

No existe el método para crear un nuevo proyecto en *Flow Simulation*.

Solución 1 (Finalmente no llevada a cabo):

No hay solución visible. La alternativa es pedir al usuario que sea él el que cree el proyecto de Flow manualmente con el modelo CAD de la aeronave deseada. De esta manera, el programa inicia su ejecución con el archivo con proyecto creado.

Problema 2:

Ante la falta de métodos que permitan introducir todos los parámetros del estudio de *Flow Simulation*, resulta complicado completar esa parte de la programación.

Solución 2:

Se decide postponer la programación de esa parte y posteriormente descartar, por la falta de recursos.

Problema 3:

Al pasar a programar en el módulo *Motion*, se decide seguir con el lenguaje VBA. Pero el problema llega cuando los archivos de ayuda proporcionados por SOLIDWORKS® no están disponibles en el lenguaje VBA.

Solución 3:

Se decide cambiar al lenguaje VB.NET, que tiene gran parecido con VBA y además es bastante más versátil. Para esto es necesario instalé otro entorno/interfaz de desarrollo (IDE: Visual Studio 2019).

Problema 4:

Al desarrollar el código e intentar compilarlo aparecen errores, principalmente conceptuales sobre programación. El mecanismo "QueryInterface" no funcionaba correctamente, por no tener en consideración el concepto de Identificador Global Único (o GUID, con las siglas en inglés). Por tanto, los objetos creados no soportaban las interfaces deseadas

Solución 4:

Decido instalar una plantilla (SWAddin) para el desarrollo de la aplicación. Esta plantilla resulta de gran ayuda porque te proporciona declaraciones, importaciones, librerías y métodos esenciales. De esta manera, se evita que se olvide o no se tenga en consideración elementos vitales en la ejecución del código.

Problema 5:

No es posible cargar un archivo de datos separados por comas (.csv) en una función del tipo Segmentos para los parámetros Fuerza o Motor.

El tipo de función expresada en segmentos no se puede seleccionar en el código como propiedad del parámetro.

Solución 5:

Como alternativa se decide cargar los datos como una spline y evitar los segmentos.

Los archivos han de estar formados por pares de datos separados por comas (,), con el punto (.) como separador decimal y sin separador de unidades de millar. Ningún otro tipo es válido.

Problema 6:

Al editar el parámetro "Velocidad inicial", no es posible modificar el valor constante del motor lineal que representa.

Solución 6:

Se introduce como un archivo de datos (.csv) que contenga tres puntos, con el mismo valor de velocidad. Así, hacemos que esta sea constante en los 0,2 segundos iniciales.

Problema 6:

Al editar el parámetro "Ángulo de ataque", se inserta un archivo (.csv) con el ángulo de ataque en cada momento del vuelo a modo de prueba. El programa carga la función, pero con unos valores del ángulo de ataque excesivamente mayores.

Solución 6:

Cambiar las unidades de grados sexagesimales a radianes, ya que SOLIDWORKS® lee los datos en radianes.

Problema 7:

Una vez establecidos todos los parámetros variables en el código, al ejecutar el programa SOLIDWORKS® no calcula correctamente la simulación. Se para justo en el momento en que el ensamblaje es sometido a algún parámetro regido por una función formada por un archivo de datos.

Solución 7:

Redondear las zonas de las funciones donde el cambio de pendiente es muy elevado.

Al proporcionar un conjunto de datos, SOLIDWORKS® permite seleccionar tres métodos de interpolación: Spline Akima, Spline cúbica o Lineal. El método de interpolación que se selecciona se utiliza para definir la función del perfil entre los puntos de datos. [17]

En los dos parámetros que se rigen por una función formada por segmentos, se escoge la interpolación lineal por ser la más parecida.

El método de interpolación lineal converge más rápidamente que los otros métodos. La función resultante es una función lineal continua que tiene una derivada discontinua en los puntos de datos que se proporcionen. La segunda derivada es cero, excepto en los puntos da datos proporcionados donde es infinita. Con el fin de evitar errores a la hora de converger será necesario suavizar las segundas derivadas.

5.Futuras líneas de desarrollo

Adición del módulo Flow Simulation

Uno de los objetivos no completados es la inclusión en la aplicación de la parte de Flow *Simulation* de SOLIDWORKS®. Ante la incapacidad de dar con varias funcionalidades: crear un nuevo proyecto, modificar las unidades, seleccionar el tipo de flujo y crear un estudio paramétrico, sumado a la escasa información al respecto se decide descartar esta fracción del proyecto.

Con una mayor comprensión del lenguaje de programación VBA o probando con otro diferente como Visual C++, se podría intentar alcanzar el proyecto completo deseado.

Simplificación del código

La consecución de un programa informático que funcione correctamente no es compleja, aunque el tiempo disponible sea escaso. Pero la escasa experiencia en este ámbito puede derivar en redundancias dentro del código, que supongan rendimientos no del todo completos.

De esta manera, siendo un amplio conocedor de las herramientas que un determinado lenguaje proporciona, sería posible remodelar el código para conseguir un funcionamiento más eficaz.

Adición de herramientas para mejorar experiencia del usuario

Poder hacer varios estudios seguidos y unir los resultados para compararlos en un único documento.

Recreación con el mejor simulador

La aplicación creada en este trabajo de final de grado ha estado basada en la simulación hecha por Francesc Llobera [1], es decir, los parámetros del vuelo parabólico han sido los mismos, dejando, eso sí, como incógnita los variables.

Siendo ahora conocedores de la existencia de trabajos de otros alumnos, en los que han podido mejorar la fiabilidad del simulador, sería interesante volver a crear esta aplicación paramétrica basándose en la mejor opción.

Parametrización del modelado de la aeronave

Teniendo en cuenta que la mayor parte de documentación e información sobre la API de SOLIDWORKS® trata sobre la automatización del modelado del sólido en 3D, podría resultar de gran utilidad parametrizar el modelado de la aeronave.

Quizá sea una propuesta desproporcionada, por la variedad de geometrías existentes en el mundo de la aeronavegación, pero se trata proceso que de otra forma requiere una inversión de trabajo importante.

6.Análisis de impacto ambiental

Uno de los aspectos positivos de este proyecto es que, con él, evitamos recurrir a las pruebas de comprobación sobre qué parábola es más eficiente durante el vuelo y, aunque sea necesario validar las predicciones calculadas, se evita descubrir el funcionamiento en pleno vuelo. Este hecho supone el ahorro de varios vuelos y, por tanto, un menor daño sobre el medio ambiente.

Si se toma como ejemplo el vuelo de una avioneta acrobática, como la estudiada en el trabajo de Francesc Llobera [2] se puede calcular, aproximadamente, las emisiones de CO_2 generadas:

(Cálculos basados en los datos de un manual de vuelo del año 1989 de la aeronave Sukhoi-26 M)

A una velocidad de crucero de 200 km/h, el consumo de la aeronave Sukhoi 26 es de 39 l/h.

Y teniendo en cuenta que un experimento de este tipo puede durar unos 30 minutos [18].

$$0{,}5\ h\ \times \frac{39\ l}{h} = 19{,}75\ litros$$

Si se añade el consumo durante el despegue y durante el aterrizaje

Despegue (a 500 m de altura)

$$3\ {}^{l}/_{km} \times 3\ km = 9\ litros$$

Aterrizaje (a 1000 m de altura)

$$0{,}6\ {}^{l}/_{km} \times 3{,}5\ km = 2{,}1\ litros$$

Si se usa el combustible *Av-gas*, el más habitual para este tipo de aeronaves, las emisiones de CO_2 son de 2,1994 kg/l [19].

Por tanto, el total de emisiones por vuelo será:

$$(19{,}75\ l + \ 9\ l + \ 2{,}1\ l)\ \times 2{,}1994\ {}^{kg}/_{l} = 67{,}85\ kg\ de\ CO_2$$

Comparándolo con un trayecto de 100 km en un automóvil de gasolina que consuma 6l/100km

$$2{,}3749\ {}^{kg}/_{l} \times \frac{6\ l}{100\ km} \times 100km = 14{,}24kg\ de\ CO_2$$

sería el equivalente a realizar un viaje de 476 km.

Por tanto, se puede afirmar que la reducción en emisiones que se va a conseguir con un simulador fiable va a ser muy importante.

Mención aparte merece el medio de transporte escogido para ir a la universidad durante el desarrollo del proyecto, que no ha sido otro que la bicicleta.

7.Análisis económico

Si se tiene en cuenta que un experimento académico en la ISS puede costar al menos 80.000$ si es en el interior de un CubeSat o 35.000$ como mínimo si es en un NanoLab [20], realizarlo durante la ejecución de varias parábolas en un vuelo con una pequeña aeronave supone una rentabilidad evidente.

Si, además, es posible tener un simulador fiable para evitar los vuelos de comprobación de la eficacia de la geometría de la parábola, el ahorro es todavía mayor.

A continuación, se muestra el desglose del presupuesto para completar el proyecto de un simulador paramétrico. Se ha de tener en cuenta que en este caso la licencia educacional de SOLIDWORKS® ha sido proporcionada por el centro.

Concepto		Precio / ud.	Cantidad	Precio (€)
Ingeniería	**Investigación documental**	30 €/h	300 h	9 000,00
	Gestión del proyecto	30 €/h	100 h	3 000,00
	Desarrollo código SOLIDWORKS	30 €/h	100 h	3 000,00
	Desarrollo código Excel	30 €/h	50 h	1 500,00
	Comunicación	30 €/h	50 h	1 500,00
Documentación (Libro técnico)		40 €/ud.	1 ud.	40,00
Licencia SOLIDWORKS® Student		100 €/ud.	1 ud.	100,00
Material papelería		20 €/ud.	1 ud.	20,00
		Subtotal		18 160,00
		IVA (21%)		3 813,60
		TOTAL		**21 973,60**

Tabla 6. Presupuesto.

8.Conclusiones

A pesar de haber tenido que descartar el módulo de *Flow Simulation* y de esta forma, no haber podido alcanzar el objetivo previsto inicialmente, que consistía en parametrizar todo el proceso de los trabajos anteriores de la línea de investigación, sí se ha podido parametrizar en una misma aplicación:

- Los cálculos posteriores a la simulación del viento en el módulo *Flow Simulation*
- La ejecución de la simulación con el módulo de *Motion*
- Los cálculos y presentación de datos finales

Se consigue reducir, de esta forma, los tiempos de un proyecto completo típico para estudiar la afección de la ingravidez. Formado, generalmente, por la preparación del experimento, análisis de vuelo, vuelo junto con los elementos para experimentación, tratamiento de datos y extracción de conclusiones.

Supondría reducir, de forma considerable, la parte en la que hay que analizar la trazada durante el vuelo, ya que si se considera a un usuario sin conocimiento previo de la herramienta de simulación desarrollada y se compara el caso previo de simulación y el actual parametrizado se obtienen las inversiones de tiempo aproximadas de las Tablas 7 y 8.

Proceso de simulación anterior	
Tarea	Tiempo invertido (minutos)
Comprensión del funcionamiento	60
Cálculos posteriores a la simulación de fluido	30
Establecimiento de relaciones de posición y adición de los parámetros en *Motion*	60
Extracción de resultados y de conclusiones	30
	180 = 3 horas

Tabla 7. Tiempo de estudio antes.

Proceso de simulación actual (parametrizado)	
Tarea	Tiempo invertido (minutos)
Instalación	5
Comprensión del funcionamiento	22
Ejecución del complemento creado	3
	30 minutos

Tabla 8. Tiempo de estudio ahora.

Aun así, sería posible mejorar estos resultados, tal y como se ha mencionado en el apartado 5. sobre futuras investigaciones; haciendo hincapié, sobre todo, en buscar una solución a la programación del módulo *Flow Simulation*..

9.Bibliografía, webgrafía y trabajos citados

1. **Brigos, M., Perez-Poch, A., Alpiste, F** (2014). Microgravity Science and Technology. *Parabolic Flights with Single-Engine Aerobatic Aircraft: Flight Profile and a Computer Simulator for its Optimization*. Vol. 26, pp. 229-239. URL: https://doi.org/10.1007/s12217-014-9382-0
2. **Llobera Herrera, F**. (2017). *Simulador de vol parabòlic amb l'aeronau acrobàtica de Shukhoi* (Trabajo Final de Grado). Universitat Politècnica de Catalunya.
3. **Muñoz Navarro, M.A.** *Manual de vuelo*. Consultado en enero de 2020. URL: https://www.manualvuelo.es/index.html
4. **May, Sandra** (2017). NASA. *What is microgravity?*. Consultado en enero de 2020 URL: https://www.nasa.gov/audience/forstudents/5-8/features/nasa-knows/what-is-microgravity-58.html
5. **Carvajal, G.** (2017). La Brujula Verde. La torre de caída libre de Bremen, un laboratorio de microgravedad único en Europa. Consultado en enero de 2020. URL: https://www.labrujulaverde.com/2017/10/la-torre-de-caida-libre-de-bremen-un-laboratorio-de-microgravedad-unico-en-europa
6. Air Zero-G. *How parabolic flights work?*. Consultado en enero de 2020. URL: https://www.airzerog.com/zero-g-flights-how-it-works/
7. **Shin, J. H.** & **Kwak, B. M.** (1999). *Integration of commercial codes from CAD to optimization for structural design.* In *Proceedings of the First Japan-Korea Joint Symposium on Optimization of Structurland Mechanical Systems* (pp. 649 – 655)
8. **SOLIDWORKS Corp**. (2018). Solidworks. *SOLIDWORKS® FlowSimulation*. URL: https://www.solidworks.com/es/product/solidworks-flow-simulation
9. *Interfaz de programación de aplicaciones*. En Wikipedia. Consultado el 5 de diciembre de 2019. URL: https://help.solidworks.com/2018/Spanish/SolidWorks/motionstudies/c_interpolation_method_comparison.htm?id=68a9023441404fd584c5aaa9d14e2a24#Pg0
10. **Dassault Systemes (2018)**. *SOLIDWORKS API Help*. URL: https://help.solidworks.com/2019/English/api/sldworksapiprogguide/Welcome.htm
11. **Dassault Systemes (2018)**. *Types of SOLIDWORKS API Applications*. URL: https://help.solidworks.com/2019/english/api/SWHelp_List.html?id=968bd8a6e720497696b7aa6514e4341b#Pg0
12. **Marc D'Aoust** (2006). *Avoid writing tedious, boring code*. https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-6.0/aa227881(v=vs.60)?redirectedfrom=MSDN
13. **Andrew Troelsen** (2008). *Pro VB 2008 and the .NET 3.5 Platform: The expert's voice in .NET*. Ed. Apress, p. 5.
14. **Code Stack** (2019). *Fix missing Visual Studio SOLIDWORKS API SDK project templates*. URL: https://www.codestack.net/solidworks-api/troubleshooting/addins/sdk-installation/
15. **Code Stack** (2019). *SOLIDWORKS API Object model class hierarchy diagram*. URL: https://www.codestack.net/solidworks-api/getting-started/api-object-model/class-diagram/
16. **Code Stack** (2019). *Creating the Visual Studio Installer (VSI) for SOLIDWORKS application*. URL:https://www.codestack.net/solidworks-api/deployment/installer/vsi/
17. **SOLIDWORKS Corp**. (2018). Ayudas de SOLIDWORKS. *Comparación entre los métodos de interpolación*. URL:

https://help.solidworks.com/2018/Spanish/SolidWorks/motionstudies/c_interpolation_method_comparison.htm?id=68a9023441404fd584c5aaa9d14e2a24#Pg0

18. **Paco Rego** (2019). *El vuelo "acrobático" que convierte 10 pajuelas de semen en "supersemen"*. Extraído del diario El Mundo. URL: https://www.elmundo.es/cronica/2019/07/26/5d31b7ccfc6c83aa458b457e.html
19. **U.S. Energy Administration** (2016). *Carbon Dioxide Emissions Coefficient*. URL: https://www.eia.gov/environment/emissions/co2_vol_mass.php
20. **Nanoracks LLC** (2020). *FAQ. What is your Price?*. URL: http://nanoracks.com/resources/faq/

ANEXOS

Volumen II

ANEXO I:

```
#Region "UI Callbacks"

Sub Calculo()
        'Tratamiento de datos pre-SOLIDWORKS Motion Simulation
        Dim ExcelApp As New Microsoft.Office.Interop.Excel.Application
        ExcelApp.Visible = True

        'Abro el documento EXCEL ya creado, que está vacio, a excepción de los títulos de las
columnas
        Dim xlsxFile As String
        xlsxFile = Form1.miFormulario.Ubicacion1 & "\Informe.xlsx"
        Dim Libro = ExcelApp.Workbooks.Open(xlsxFile)

        'Creo la Hoja2 (después de la Hoja1)
        Libro.Sheets.Add(After:=Libro.Sheets(1))
        'Creo la Hoja3 (después de la Hoja2)
        Libro.Sheets.Add(After:=Libro.Sheets(2))

        'Creo una lista con todos los ángulos de ataque
        Dim Angulos As New List(Of Object) From
            {Form1.miFormulario.Angulo1, Form1.miFormulario.Angulo2, Form1.miFormulario.Angulo3,
Form1.miFormulario.Angulo4, Form1.miFormulario.Angulo5, Form1.miFormulario.Angulo6,
Form1.miFormulario.Angulo7, Form1.miFormulario.Angulo8, Form1.miFormulario.Angulo9,
Form1.miFormulario.Angulo10}
        Dim i As Integer
        Dim ACount As Integer

        ACount = 0
        For i = 0 To Angulos.Count - 1
            If Angulos(i).Text IsNot String.Empty Then
                ACount = i + 1
            ElseIf Angulos(i).Text = String.Empty Then
                Exit For
            End If
        Next i

        'Creo una lista con todas las velocidades
        Dim Velocidades As New List(Of Object) From
            {Form1.miFormulario.Velocidad1, Form1.miFormulario.Velocidad2,
Form1.miFormulario.Velocidad3, Form1.miFormulario.Velocidad4, Form1.miFormulario.Velocidad5,
Form1.miFormulario.Velocidad6, Form1.miFormulario.Velocidad7, Form1.miFormulario.Velocidad8,
Form1.miFormulario.Velocidad9, Form1.miFormulario.Velocidad10}
        Dim VCount As Integer

        VCount = 0
        For i = 0 To Velocidades.Count - 1
            If Velocidades(i).Text IsNot String.Empty Then
                VCount = i + 1
            ElseIf Velocidades(i).Text = String.Empty Then
                Exit For
            End If
        Next i


        'Traslado los ángulos y velocidades desde el formulario de entrada de datos al Libro de
Excel
        ' también los paso a radianes y m/s respectivamente
        Dim Total As Integer
        Total = ACount * VCount
        Dim j, k, l As Integer
        j = 0
```

```
        k = 1
        l = 0

        For i = 6 To Total + 5
            Libro.Sheets(1).Cells(i, 1) = i - 5
            Libro.Sheets(1).Cells(i, 2) = Val(Angulos(j).Text)
            Libro.Sheets(1).Cells(i, 3) = Val(Velocidades(l).Text)
            Libro.Sheets(1).Cells(i, 4) = Libro.Sheets(1).Cells(i, 2).Value * System.Math.PI / 180
            Libro.Sheets(1).Cells(i, 5) = Libro.Sheets(1).Cells(i, 3).Value * 1000 / 3600

            l = l + 1
            If i - 5 = VCount * k Then
                j = j + 1
                k = k + 1
                l = 0
            End If

        Next i


        'Inserto en el Libro de Excel los datos de Flow, tomando como origen la ruta del .csv examinado
        Dim Hoja1 As Worksheet
        Hoja1 = Libro.Sheets("Hoja1")
        With Hoja1.QueryTables.Add(Connection:="TEXT;" & Form1.miFormulario.Label7.Text, Destination:=Hoja1.Range("F6"))
            .TextFileParseType = 1
            .TextFileCommaDelimiter = True
            .TextFileDecimalSeparator = "."
            .TextFileThousandsSeparator = " "
            .Refresh()
        End With


        'Modifico el archivo .csv con la velocidad inicial proporcionada por el usuario
        Dim VelocidadInicialFile As String
        VelocidadInicialFile = Form1.miFormulario.Ubicacion1 & "\VelocidadInicial.csv"
        Dim Reemplazo As String = File.ReadAllText(VelocidadInicialFile)
        Reemplazo = Reemplazo.Replace("90", Form1.miFormulario.VelocidadInicial.Text)
        File.WriteAllText(VelocidadInicialFile, Reemplazo)


        'Hago los cálculos para poder graficar los polinomios de los Coeficientes de Sustentación y Resistencia
        Dim ValorY, ValorZ As Double
        For i = 6 To Total + 5
            ValorY = Libro.Sheets(1).Cells(i, 6).Value
            ValorZ = Libro.Sheets(1).Cells(i, 7).Value

            Libro.Sheets(1).Cells(i, 9) = System.Math.Cos(Libro.Sheets(1).Cells(i, 4).Value) * (Abs(ValorY)) -
                System.Math.Sin(Libro.Sheets(1).Cells(i, 4).Value) * (Abs(ValorZ))
            Libro.Sheets(1).Cells(i, 10) = System.Math.Sin(Libro.Sheets(1).Cells(i, 4).Value) * (Abs(ValorY)) +
                System.Math.Cos(Libro.Sheets(1).Cells(i, 4).Value) * (Abs(ValorZ))
        Next i

        For i = 6 To Total + 5
            Libro.Sheets(1).Cells(i, 12) = Libro.Sheets(1).Cells(i, 9).Value / (Libro.Sheets(1).Cells(i, 5).Value ^ 2)
            Libro.Sheets(1).Cells(i, 14) = Libro.Sheets(1).Cells(i, 10).Value / (Libro.Sheets(1).Cells(i, 5).Value ^ 2)
        Next i
        j = 0
        k = 1
        For i = 6 To Total + 5
```

```
            If i - 5 = VCount * k Then
                Dim Rango As Range
                Dim Celda1, Celda2 As String
                Dim CoordenadaCelda1 As Integer = (i - 5)
                Dim CoordenadaCelda2 As Integer = (i)
                Celda1 = CoordenadaCelda1.ToString()
                Celda2 = CoordenadaCelda2.ToString()
                Rango = Libro.Worksheets("Hoja1").Range("L" & Celda1 & ":L" & Celda2)
                Libro.Sheets(1).Cells(i - 5, 13) = ExcelApp.WorksheetFunction.Average(Rango)
                Libro.Sheets(1).Range("M" & Celda1 & ":M" & Celda2).Merge

                Rango = Libro.Worksheets("Hoja1").Range("N" & Celda1 & ":N" & Celda2)
                Libro.Sheets(1).Cells(i - 5, 15) = ExcelApp.WorksheetFunction.Average(Rango)
                Libro.Sheets(1).Range("O" & Celda1 & ":O" & Celda2).Merge

                Libro.Sheets(1).Cells(i - 5, 16) = Val(Angulos(j).Text)
                Libro.Sheets(1).Range("P" & Celda1 & ":P" & Celda2).Merge
                j = j + 1
                k = k + 1
            End If
        Next i


        'Hago los gráficos en los que se representan los Coeficientes de Sustentación y Resistencia
VS ángulo de ataque
        Dim Graficos1 As ChartObject
        Dim Grafico1 As ChartObjects
        Dim Serie1 As Series
        Dim Ecuacion1 As String
        Dim LineaTendencia1 As Trendline
        Dim CoordenadaCeldaFinal As Integer = Total + 5
        Dim CeldaFinal As String = CoordenadaCeldaFinal.ToString()
        Dim RangoX1 As Range = Libro.Worksheets("Hoja1").Range("P6:P" & CeldaFinal)
        Dim RangoY1 As Range = Libro.Worksheets("Hoja1").Range("M6:M" & CeldaFinal)

        Grafico1 = Libro.Worksheets(1).ChartObjects
        Graficos1 = Grafico1.Add(Left:=940, Width:=425, Top:=50, Height:=283).Activate()

        Libro.Worksheets(1).ChartObjects(1).Chart.ChartType = XlChartType.xlXYScatter
        Libro.Worksheets(1).ChartObjects(1).Chart.HasTitle = True
        Libro.Worksheets(1).ChartObjects(1).Chart.ChartTitle.Text = "Coef.  Resistencia VS Ángulo de
ataque"
        Libro.Worksheets(1).ChartObjects(1).Chart.Axes(XlAxisType.xlCategory).HasTitle = True
        Libro.Worksheets(1).ChartObjects(1).Chart.Axes(XlAxisType.xlCategory).AxisTitle.Text =
"Ángulo de ataque (grados)"
        Libro.Worksheets(1).ChartObjects(1).Chart.Axes(XlAxisType.xlValue).HasTitle = True
        Libro.Worksheets(1).ChartObjects(1).Chart.Axes(XlAxisType.xlValue).AxisTitle.Text =
"Coef.Resistencia"

        Serie1 = Libro.Worksheets(1).ChartObjects(1).Chart.SeriesCollection.NewSeries()
        Serie1.XValues = RangoX1
        Serie1.Values = RangoY1
        LineaTendencia1 = Serie1.Trendlines.Add(Type:=3, Order:=4, DisplayEquation:=True)


        Dim Graficos2 As ChartObject
        Dim Grafico2 As ChartObjects
        Dim Serie2 As Series
        Dim Ecuacion2 As String
        Dim LineaTendencia2 As Trendline
        Dim RangoX2 As Range = Libro.Worksheets("Hoja1").Range("P6:P" & CeldaFinal)
        Dim RangoY2 As Range = Libro.Worksheets("Hoja1").Range("O6:O" & CeldaFinal)

        Grafico2 = Libro.Worksheets(1).ChartObjects
```

```
        Graficos2 = Grafico2.Add(Left:=940, Width:=425, Top:=343, Height:=283).Activate()

        Libro.Worksheets(1).ChartObjects(2).Chart.ChartType = XlChartType.xlXYScatter
        Libro.Worksheets(1).ChartObjects(2).Chart.HasTitle = True
        Libro.Worksheets(1).ChartObjects(2).Chart.ChartTitle.Text = "Coef.  Resistencia VS Ángulo de
ataque"
        Libro.Worksheets(1).ChartObjects(2).Chart.Axes(XlAxisType.xlCategory).HasTitle = True
        Libro.Worksheets(1).ChartObjects(2).Chart.Axes(XlAxisType.xlCategory).AxisTitle.Text =
"Ángulo de ataque (grados)"
        Libro.Worksheets(1).ChartObjects(2).Chart.Axes(XlAxisType.xlValue).HasTitle = True
        Libro.Worksheets(1).ChartObjects(2).Chart.Axes(XlAxisType.xlValue).AxisTitle.Text =
"Coef.Resistencia"

        Serie2 = Libro.Worksheets(1).ChartObjects(2).Chart.SeriesCollection.NewSeries()
        Serie2.XValues = RangoX2
        Serie2.Values = RangoY2
        LineaTendencia2 = Serie2.Trendlines.Add(Type:=3, Order:=4, DisplayEquation:=True)


        ' Obtengo los polinomios de ambas líneas de tendencia
        Ecuacion1 = LineaTendencia1.DataLabel.Text
        Dim Caracteres1 As Integer ' = Ecuacion1.Count
        Caracteres1 = Ecuacion1.Length
        Dim y As String = "y = "
        Dim x4 As String = "x4"
        Dim x3 As String = "x3"
        Dim x2 As String = "x2"
        Dim x As String = "x "
        Dim Sustitucion1 As String = "(("
        Dim Sustitucion2 As String = "*{Desplazamiento angular5}^4"
        Dim Sustitucion3 As String = "*{Desplazamiento angular5}^3"
        Dim Sustitucion4 As String = "*{Desplazamiento angular5}^2"
        Dim Sustitucion5 As String = "*{Desplazamiento angular5}"

        Dim ExpressionS1, ExpressionS2, ExpressionS3, ExpressionS4, ExpressionS5,
ExpressionSustentacion As String
        ExpressionS1 = Ecuacion1.Replace(y, Sustitucion1)
        ExpressionS2 = ExpressionS1.Replace(x4, Sustitucion2)
        ExpressionS3 = ExpressionS2.Replace(x3, Sustitucion3)
        ExpressionS4 = ExpressionS3.Replace(x2, Sustitucion4)
        ExpressionS5 = ExpressionS4.Replace(x, Sustitucion5)
        ExpressionSustentacion = ExpressionS5 & ")*{Velocidad lineal5}*{Velocidad lineal5})"


        Ecuacion2 = LineaTendencia2.DataLabel.Text
        Dim Caracteres2 As Integer ' = Ecuacion2.Count
        Caracteres2 = Ecuacion2.Length
        Dim ExpressionR1, ExpressionR2, ExpressionR3, ExpressionR4, ExpressionR5,
ExpressionResistencia As String
        ExpressionR1 = Ecuacion2.Replace(y, Sustitucion1)
        ExpressionR2 = ExpressionR1.Replace(x4, Sustitucion2)
        ExpressionR3 = ExpressionR2.Replace(x3, Sustitucion3)
        ExpressionR4 = ExpressionR3.Replace(x2, Sustitucion4)
        ExpressionR5 = ExpressionR4.Replace(x, Sustitucion5)

        ExpressionResistencia = ExpressionR5 & ")*{Velocidad lineal5}*{Velocidad lineal5})"



        ExcelApp.ActiveWindow.Activate()


        'Comienzo el proceso SOLIDWORKS
        Dim swApp As SldWorks
        swApp = CreateObject("SldWorks.Application")
```

```
243         swApp.Visible = True
244
245         'Abrir el modelo Motion_Ignacio.SLDASM
246         Dim RutaEnsamblaje As System.String
247         Dim swModel As ModelDoc2
248         Dim swModelDocExt As ModelDocExtension
249
250         RutaEnsamblaje = Form1.miFormulario.Ubicacion1 & "\SimuladorVueloParam.SLDASM"
251         swModel = swApp.OpenDoc6(RutaEnsamblaje, 2, 1, "", 1, 1)
252         swModelDocExt = swModel.Extension
253
254         'Obtener MotionManager
255         Dim swMotionMgr As MotionStudyManager
256
257         swMotionMgr = swModelDocExt.GetMotionStudyManager()
258         If (swMotionMgr Is Nothing) Then
259             Return
260         End If
261
262         'Obtener el estudio de motion llamado: "Su26_1"
263         Dim MotionStudyName As System.String
264         Dim swMotionStudy1 As MotionStudy
265
266         MotionStudyName = "Su26_1"
267         swMotionStudy1 = swMotionMgr.GetMotionStudy(MotionStudyName)
268         If (swMotionStudy1 Is Nothing) Then
269             MsgBox("Motion Study 1 is not available.")
270             Return
271         End If
272
273         Dim boolstatus As Boolean
274
275         '¿Está el estudio de motion activado?  Si no lo está, activarlo.
276         If Not swMotionStudy1.Activate() Then swMotionStudy1.Activate()
277
278         MessageBox.Show("¿Desea continuar con la simulación?", "", MessageBoxButtons.OK,
279 MessageBoxIcon.Information)
280
281         'Modifico los features de SOLIDWORKS Motion según los datos entrados por el usuario
282         Dim swFeature As Feature
283         Dim FeatureName As String
284         Dim motionFeatures As Object
285
286         'FUERZA2 - Sustentación - VARIABLE [¡FUNCIONA!]
287         Dim Fuerza2 As String
288         Dim Sustentacio As IFeature
289         Dim SustentacioFeature As ISimulationForceFeatureData
290         Dim FuncionSustentacio As String
291
292         Fuerza2 = "Sustentacio"
293         motionFeatures = swMotionStudy1.GetMotionFeatures()
294
295         For i = 0 To UBound(motionFeatures)
296             swFeature = motionFeatures(i)
297             FeatureName = swFeature.Name
298             If FeatureName = Fuerza2 Then
299                 Sustentacio = swFeature
300                 SustentacioFeature = Sustentacio.GetDefinition()
301                 FuncionSustentacio = SustentacioFeature.FunctionExpression
302                 FuncionSustentacio = ExpressionSustentacion
303                 SustentacioFeature.FunctionExpression = FuncionSustentacio
304             End If
305         Next i
306
307         'FUERZA3 - Impuls SEGMENTOS PRUEBA - VARIABLE
```

```
308         Dim Fuerza3 As String
309         Dim Impuls As IFeature
310         Dim ImpulsFeature As ISimulationForceFeatureData
311
312         Fuerza3 = "Impuls"
313         motionFeatures = swMotionStudy1.GetMotionFeatures()
314
315         For i = 0 To UBound(motionFeatures)
316             swFeature = motionFeatures(i)
317             FeatureName = swFeature.Name
318             If FeatureName = Fuerza3 Then
319                 Impuls = swFeature
320                 ImpulsFeature = Impuls.GetDefinition()
321                 boolstatus = ImpulsFeature.LoadSplineData(Form1.miFormulario.Label8.Text)
322             End If
323         Next i
324
325         'MOTOR 3 - Velocidad Inicial - VARIABLE
326         Dim Motor3 As String
327         Dim VelocidadInicial As IFeature
328         Dim VelocitatInicialFeature As ISimulationMotorFeatureData
329
330         Motor3 = "VelocitatInicial"
331         motionFeatures = swMotionStudy1.GetMotionFeatures()
332
333         For i = 0 To UBound(motionFeatures)
334             swFeature = motionFeatures(i)
335             FeatureName = swFeature.Name
336             If FeatureName = Motor3 Then
337                 VelocidadInicial = swFeature
338                 VelocitatInicialFeature = VelocidadInicial.GetDefinition()
339                 boolstatus = VelocitatInicialFeature.LoadSplineData("VelocidadInicial.csv")
340             End If
341         Next i
342
343         'MOTOR 4 - Ángulo Ataque - VARIABLE
344         Dim Motor4 As String
345         Dim AnguloAtaque As IFeature
346         Dim AnguloAtaqueFeature As ISimulationMotorFeatureData
347
348         Motor4 = "AngAtac"
349         motionFeatures = swMotionStudy1.GetMotionFeatures()
350
351         For i = 0 To UBound(motionFeatures)
352             swFeature = motionFeatures(i)
353             FeatureName = swFeature.Name
354             If FeatureName = Motor4 Then
355                 AnguloAtaque = swFeature
356                 AnguloAtaqueFeature = AnguloAtaque.GetDefinition()
357                 boolstatus = AnguloAtaqueFeature.LoadSplineData(Form1.miFormulario.Label9.Text)
358             End If
359         Next i
360
361         'FUERZA 4 - Resistencia - VARIABLE
362         Dim Fuerza4 As String
363         Dim Resistencia As IFeature
364         Dim ResistenciaFeature As ISimulationForceFeatureData
365         Dim FuncionResistencia As String
366
367         Fuerza4 = "Resistencia"
368         motionFeatures = swMotionStudy1.GetMotionFeatures()
369
370         For i = 0 To UBound(motionFeatures)
371             swFeature = motionFeatures(i)
372             FeatureName = swFeature.Name
373             If FeatureName = Fuerza4 Then
```

```
                Resistencia = swFeature
                ResistenciaFeature = Resistencia.GetDefinition()
                FuncionResistencia = ResistenciaFeature.FunctionExpression
                FuncionResistencia = ExpressionResistencia
                ResistenciaFeature.FunctionExpression = FuncionResistencia
            End If
        Next i

        swMotionStudy1.Calculate()


        'RESULTADOS Y TRAZADOS
        'Exporto los resultados que ha generado la simulación de SOLIDWORKS
        Dim Plot1 As String
        Dim Plot2 As String
        Dim Resultados As CosmosMotionStudyResults
        Dim PlotFeatures As Object
        Dim FuerzaReaccionTotal As Feature
        Dim AlturaOrigen As Feature

        Plot1 = "FuerzaReaccionTotal"
        Plot2 = "AlturaOrigen"
        Resultados = swMotionStudy1.GetResults(4)
        PlotFeatures = Resultados.GetPlotFeatures()

        For i = 0 To UBound(PlotFeatures)
            swFeature = PlotFeatures(i)
            FeatureName = swFeature.Name
            If FeatureName = Plot1 Then
                FuerzaReaccionTotal = swFeature
                boolstatus = Resultados.ExportCSVFile(FuerzaReaccionTotal,
Form1.miFormulario.Ubicacion1 & "\Plot1_FRT.csv")
            ElseIf FeatureName = Plot2 Then
                AlturaOrigen = swFeature
                boolstatus = Resultados.ExportCSVFile(AlturaOrigen, Form1.miFormulario.Ubicacion1 &
"\Plot2_AO.csv")
            End If
        Next

        'Tratamiento de datos en Excel post-SOLIDWORKS Motion Simulation
        'Abro los .csv que ha generado SOLIDWORKS y los coloco en mi libro "Informe" de Excel
        Dim csvFile1 As String
        Dim csvFile2 As String
        csvFile1 = Form1.miFormulario.Ubicacion1 & "\Plot1_FRT.csv"
        csvFile2 = Form1.miFormulario.Ubicacion1 & "\Plot2_AO.csv"

        Dim Hoja2 As Worksheet
        Dim Hoja3 As Worksheet
        Hoja2 = Libro.Sheets("Hoja2") 'set to current worksheet name
        Hoja3 = Libro.Sheets("Hoja3")

        With Hoja2.QueryTables.Add(Connection:="TEXT;" & csvFile1, Destination:=Hoja2.Range("A1"))
            .TextFileParseType = 1
            .TextFileCommaDelimiter = True
            .TextFileDecimalSeparator = "."
            .TextFileThousandsSeparator = " "
            .Refresh()
        End With


        With Hoja3.QueryTables.Add(Connection:="TEXT;" & csvFile2, Destination:=Hoja3.Range("A1"))
            .TextFileParseType = 1
            .TextFileCommaDelimiter = True
            .TextFileDecimalSeparator = "."
            .TextFileThousandsSeparator = " "
            .Refresh()
```

```
439         End With
440
441
442         Dim PosicionRowFRT As Integer = 3
443         Dim ValorFRT As Integer = 0
444         While (Libro.Sheets(2).Range("A" & PosicionRowFRT).Value IsNot Nothing)
445             PosicionRowFRT = PosicionRowFRT + 1
446             ValorFRT = ValorFRT + 1
447         End While
448
449
450         For i = 3 To PosicionRowFRT - 1
451             Dim AceleracionG As Double = 9.807
452             Dim gForce As Double = Libro.Sheets(2).Cells(i, 2).Value
453             Libro.Sheets(2).Cells(i, 3) = (gForce / (Val(Form1.miFormulario.MasaAeronave.Text))) *
454 AceleracionG
455         Next i
456
457
458         Dim PosicionRowAO As Integer = 3
459         Dim ValorAO As Integer = 0
460         While (Libro.Sheets(3).Range("A" & PosicionRowAO).Value IsNot Nothing)
461             PosicionRowAO = PosicionRowAO + 1
462             ValorAO = ValorAO + 1
463         End While
464
465         'Grafico los valores generados por el simulador de SOLIDWORKS
466         Dim Graficos3 As ChartObject
467         Dim Grafico3 As ChartObjects
468         Dim Serie3 As Series
469         Dim RangoX3 As Range = Libro.Worksheets("Hoja2").Range("A3:A" & PosicionRowFRT)
470         Dim RangoY3 As Range = Libro.Worksheets("Hoja2").Range("C3:C" & PosicionRowFRT)
471
472         Grafico3 = Libro.Worksheets(1).ChartObjects
473         Graficos3 = Grafico3.Add(Left:=1370, Width:=425, Top:=50, Height:=283).Activate()
474
475         Libro.Worksheets(1).ChartObjects(3).Chart.ChartType = XlChartType.xlXYScatterLines
476         Libro.Worksheets(1).ChartObjects(3).Chart.HasTitle = True
477         Libro.Worksheets(1).ChartObjects(3).Chart.ChartTitle.Text = "Gravedad VS Tiempo"
478         Libro.Worksheets(1).ChartObjects(3).Chart.Axes(XlAxisType.xlCategory).HasTitle = True
479         Libro.Worksheets(1).ChartObjects(3).Chart.Axes(XlAxisType.xlCategory).AxisTitle.Text =
480 "Tiempo (segundos)"
481         Libro.Worksheets(1).ChartObjects(3).Chart.Axes(XlAxisType.xlValue).HasTitle = True
482         Libro.Worksheets(1).ChartObjects(3).Chart.Axes(XlAxisType.xlValue).AxisTitle.Text =
483 "Gravedad (g)"
484
485         Serie3 = Libro.Worksheets(1).ChartObjects(3).Chart.SeriesCollection.NewSeries()
486         Serie3.XValues = RangoX3
487         Serie3.Values = RangoY3
488
489
490         Dim Graficos4 As ChartObject
491         Dim Grafico4 As ChartObjects
492         Dim Serie4 As Series
493         Dim RangoX4 As Range = Libro.Worksheets("Hoja3").Range("A1:A" & PosicionRowAO)
494         Dim RangoY4 As Range = Libro.Worksheets("Hoja3").Range("B1:B" & PosicionRowAO)
495
496         Grafico4 = Libro.Worksheets(1).ChartObjects
497         Graficos4 = Grafico4.Add(Left:=1370, Width:=425, Top:=343, Height:=283).Activate()
498
499         Libro.Worksheets(1).ChartObjects(4).Chart.ChartType = XlChartType.xlXYScatterLines
500         Libro.Worksheets(1).ChartObjects(4).Chart.HasTitle = True
501         Libro.Worksheets(1).ChartObjects(4).Chart.ChartTitle.Text = "Altura VS Tiempo"
502         Libro.Worksheets(1).ChartObjects(4).Chart.Axes(XlAxisType.xlCategory).HasTitle = True
503         Libro.Worksheets(1).ChartObjects(4).Chart.Axes(XlAxisType.xlCategory).AxisTitle.Text =
504 "Tiempo (segundos)"
```

```
505         Libro.Worksheets(1).ChartObjects(4).Chart.Axes(XlAxisType.xlValue).HasTitle = True
506         Libro.Worksheets(1).ChartObjects(4).Chart.Axes(XlAxisType.xlValue).AxisTitle.Text = "Altura
507 respecto origen de parábola (m)"
508
509         Serie4 = Libro.Worksheets(1).ChartObjects(4).Chart.SeriesCollection.NewSeries()
510         Serie4.XValues = RangoX4
511         Serie4.Values = RangoY4
512
513         System.IO.File.WriteAllText(Form1.miFormulario.Ubicacion1 & "\Plot1_FRT.csv", "")
514         System.IO.File.WriteAllText(Form1.miFormulario.Ubicacion1 & "\Plot2_AO.csv", "")
515
516         'Cierra SOLIDWORKS sin guardar el ensamblaje para evitar que este se modifique. Sino, ¡se
517 puede producir error en futuras ejecuciones!
518         swApp.CloseAllDocuments(True)
519
520     End Sub
521
522     Sub ZeroGMotionSimulation()
523         Dim Formulario As Form1
524         Formulario = New Form1()
525         Formulario.Show()
526         Formulario = Nothing
527
528     End Sub
```

Manual de usuario

Volumen III

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

Índice

Requisitos

La presente aplicación ha sido desarrollada y probada con una máquina con las siguientes propiedades:

Procesador: Intel® Core™ i7-8550U CPU 1.80 GHz – 1.99 GHz

Memoria instalada (RAM): 16,0 GB

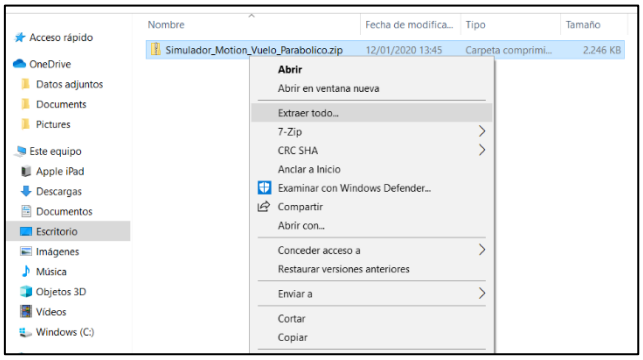
Tipo de sistema: Sistema operativo de 64 bits.

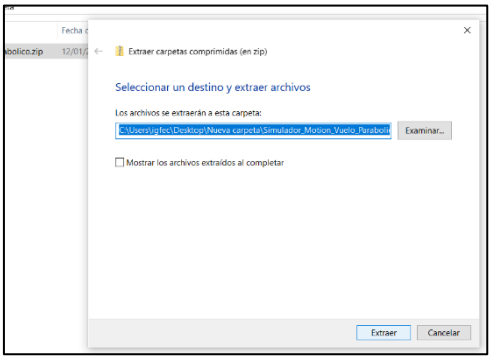
La versión de SOLIDWORKS® utilizada ha sido la 27.2.0.51 del año 2019-2020

No se garantiza el buen funcionamiento con versiones inferiores y procesadores menos potentes.

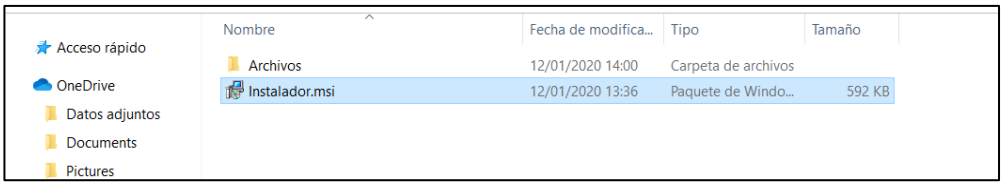
Instalación

- Lo primero que debe hacer es descomprimir la carpeta **Simulador_Motion_Vuelo_Parabolico.zip**

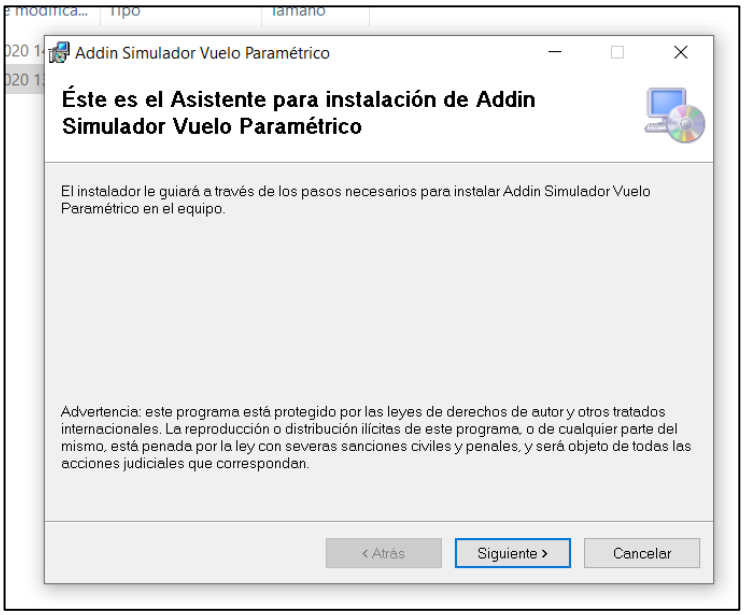




- Dentro de la carpeta **Simulador_Motion_Vuelo_Parabolico.zip** encontrará una carpeta denominada **Archivos**, que es fundamental para el buen funcionamiento del complemento y que se explica en el manual de instrucciones, y el **Instalador.msi**.



- Ejecute en instalador y siga los pasos que le indique el asistente.

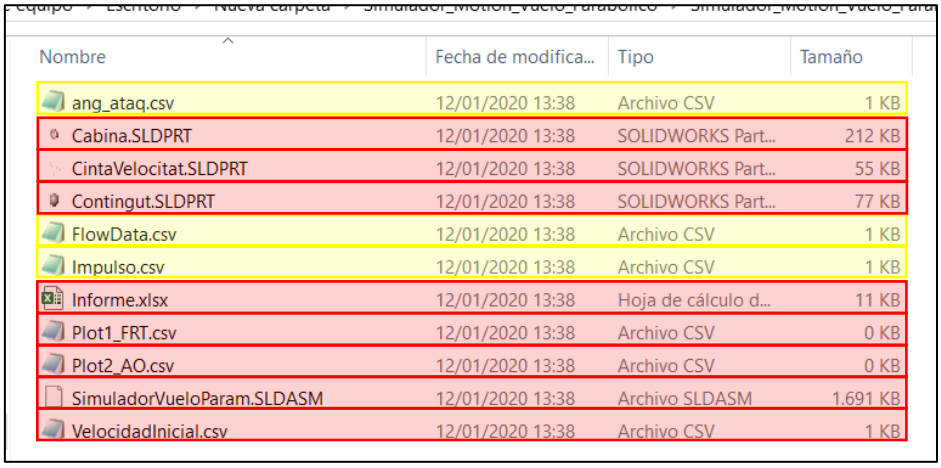


Manual de instrucciones

La carpeta Archivos es vital porque contiene todos los elementos necesarios para llevar a cabo la simulación.

Por una parte, están los elementos de SOLIDWORKS® y por otra los datos de entrada. Asimismo, hay un Libro de Excel vacío que tomará el programa para crear el informe final con los resultados obtenidos de la simulación.

De los 11 elementos presentes, solo habrá que modificar 3 de ellos: **ang_ataq.csv**, **FlowData.csv** y **Impulso.csv**



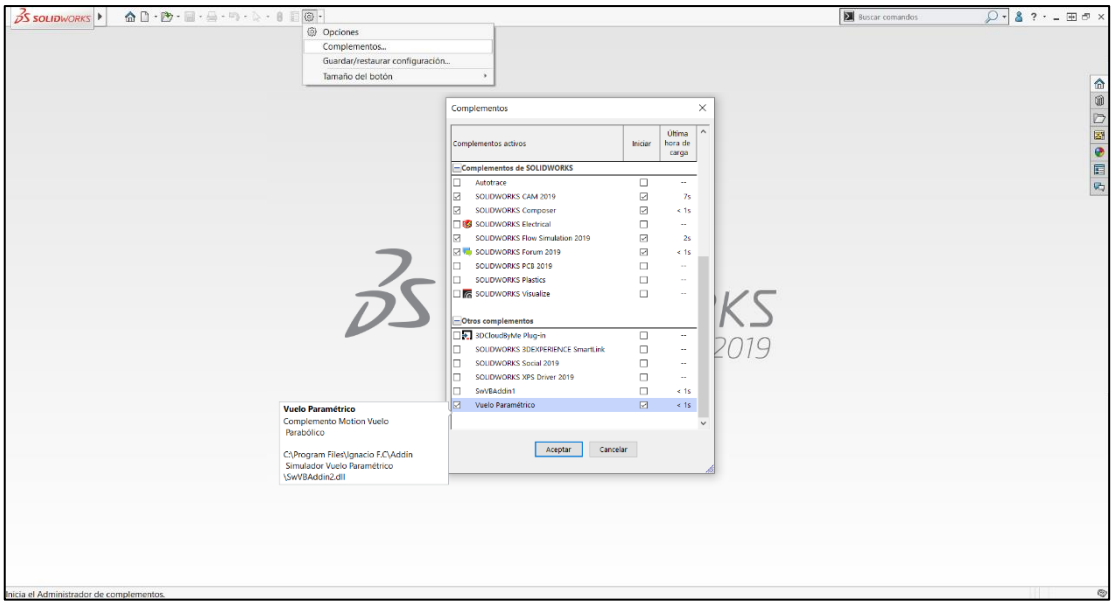
Nombre	Fecha de modifica...	Tipo	Tamaño
ang_ataq.csv	12/01/2020 13:38	Archivo CSV	1 KB
Cabina.SLDPRT	12/01/2020 13:38	SOLIDWORKS Part...	212 KB
CintaVelocitat.SLDPRT	12/01/2020 13:38	SOLIDWORKS Part...	55 KB
Contingut.SLDPRT	12/01/2020 13:38	SOLIDWORKS Part...	77 KB
FlowData.csv	12/01/2020 13:38	Archivo CSV	1 KB
Impulso.csv	12/01/2020 13:38	Archivo CSV	1 KB
Informe.xlsx	12/01/2020 13:38	Hoja de cálculo d...	11 KB
Plot1_FRT.csv	12/01/2020 13:38	Archivo CSV	0 KB
Plot2_AO.csv	12/01/2020 13:38	Archivo CSV	0 KB
SimuladorVueloParam.SLDASM	12/01/2020 13:38	Archivo SLDASM	1.691 KB
VelocidadInicial.csv	12/01/2020 13:38	Archivo CSV	1 KB

El resto de archivos han de permanecer sin cambios de ningún tipo, de lo contrario, el programa puede fallar.

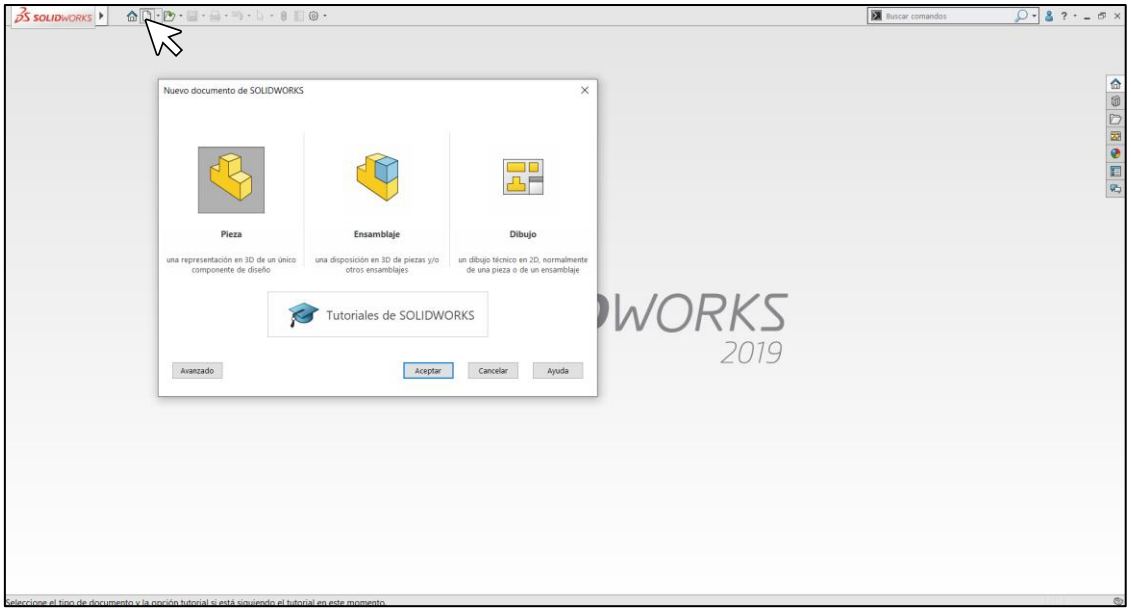
Si ya ha instalado el complemento, en la pantalla de inicio de SOLIDWORKS®, podrá ver en que ya lo tiene disponible en la pestaña **Opciones** → **Complementos**.

Actívelo ☑ y clique en **Aceptar** antes de abrir un documento en SOLIDWORKS®. Active también

en caso de que no lo esté.



En primer lugar, cree una pieza nueva.

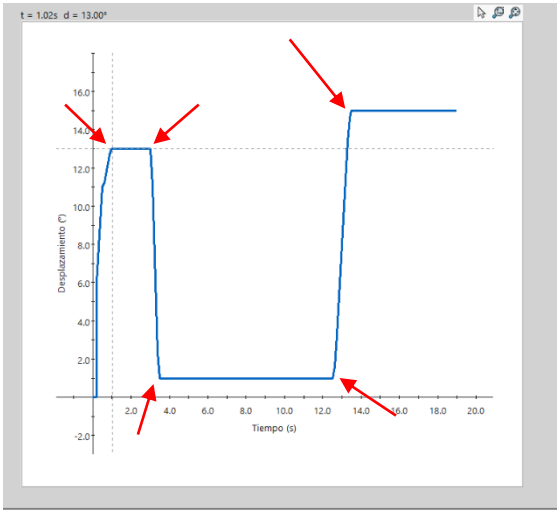


Antes de ejecutar el complemento, volvamos a los archivos que era necesario modificar.

<u>Introducciones de datos necesarias antes de ejecutar la simulación</u>

IMPORTANTE:

- Todos los archivos de datos han de tener el punto (.) como separador decimal y no han de tener ningún tipo de separador de miles.
- Cada línea contendrá un par de datos separados por una coma (,).
- Es vital, que las unidades de las magnitudes físicas sean las que se indican a continuación.
- Únicamente puede haber datos numéricos, nunca títulos encabezando.
- Al proporcionar un conjunto de datos, en **ang_ataq.csv** y en **Impulso.csv**, es muy importante que las derivadas de la función sean suaves (continuas). Para ello, los puntos introducidos no deben generar cambios de pendiente muy bruscos en el perfil de la función.

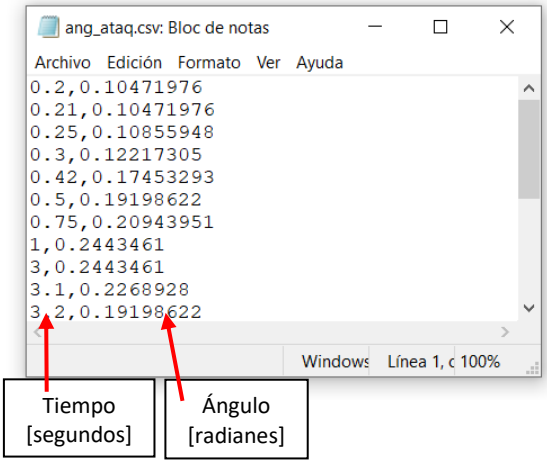


Es decir, habrá que “redondear” puntos como los señalados en la gráfica.
*Explicación completa en la memoria.

Si estas restricciones no se cumplen, el programa puede fallar.

- **ang_ataq.csv**

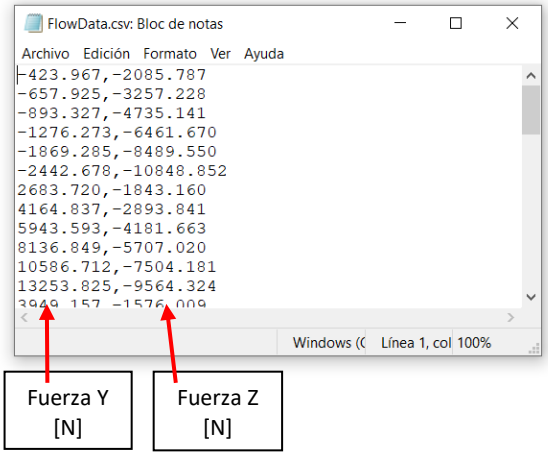
  En este archivo se han de incluir los diversos ángulos de ataque a los que se ve sometida la aeronave durante los 16 segundos de vuelo parabólico.



Cada línea tendrá como primer dato el **tiempo** [segundos] y como segundo dato el **ángulo de ataque** [radianes]

- **FlowData.csv**

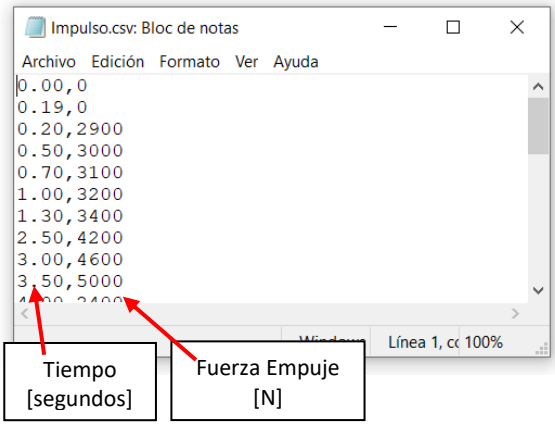
  En este archivo se han de incluir las fuerzas en el eje Z y en el eje Y, que provienen del estudio de viento realizado con SOLIDWORKS® *Flow Simulation*.



Por ejemplo, si se ha hecho un estudio con 5 ángulos de ataque diferentes para 5 velocidades diferentes, tendrá que haber 5x5=25 soluciones de Fuerza Y, Fuerza Z. Es decir, habrá 25 líneas con la **Fuerza Y** [N] como primer dato y con la **Fuerza Z** [N] como segundo dato.

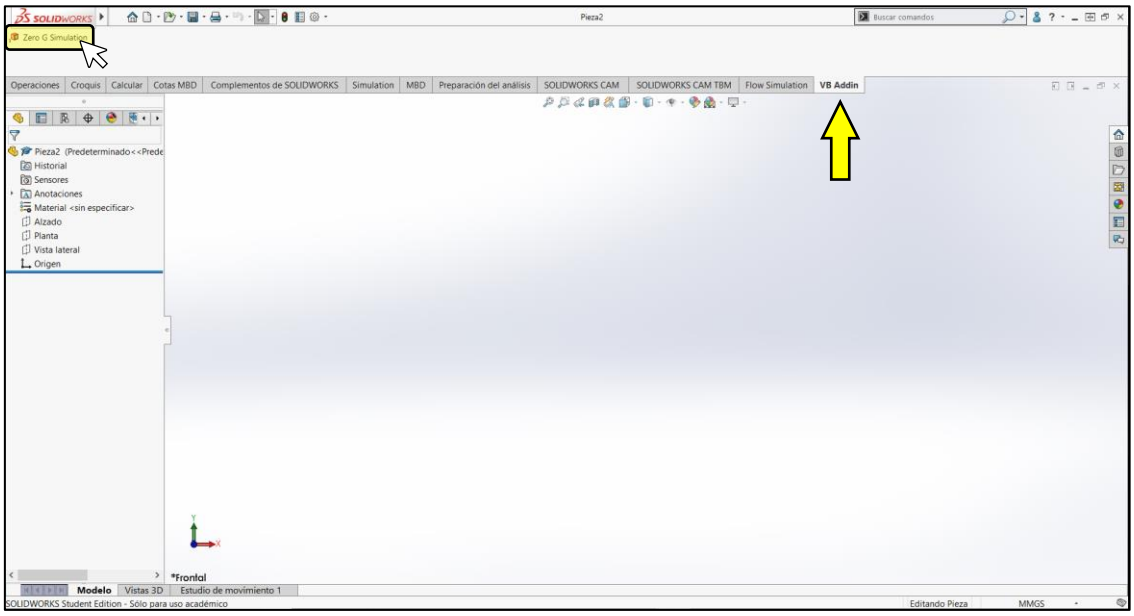
- **Impulso.csv**

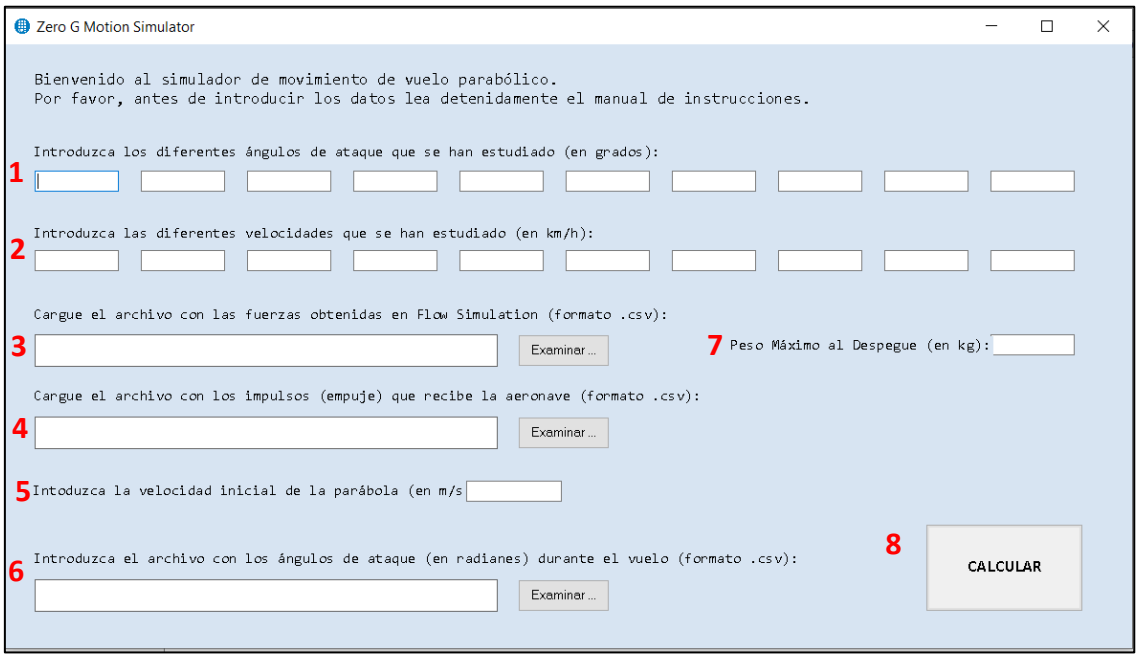
  En este archivo se han de incluir los impulsos, también conocidos como fuerza de empuje, que sufre la aeronave durante los 16 segundos de vuelo parabólico.



Cada línea tendrá como primer dato el **tiempo** [segundos] y como segundo dato el **impulso (fuerza de empuje)** [N].

Como podrá comprobar, a partir de ahora tiene disponible la pestaña **VBAddin** con el complemento instalado. Clique en el botón **Zero G Simulation** para abrir el formulario de entrada de datos.



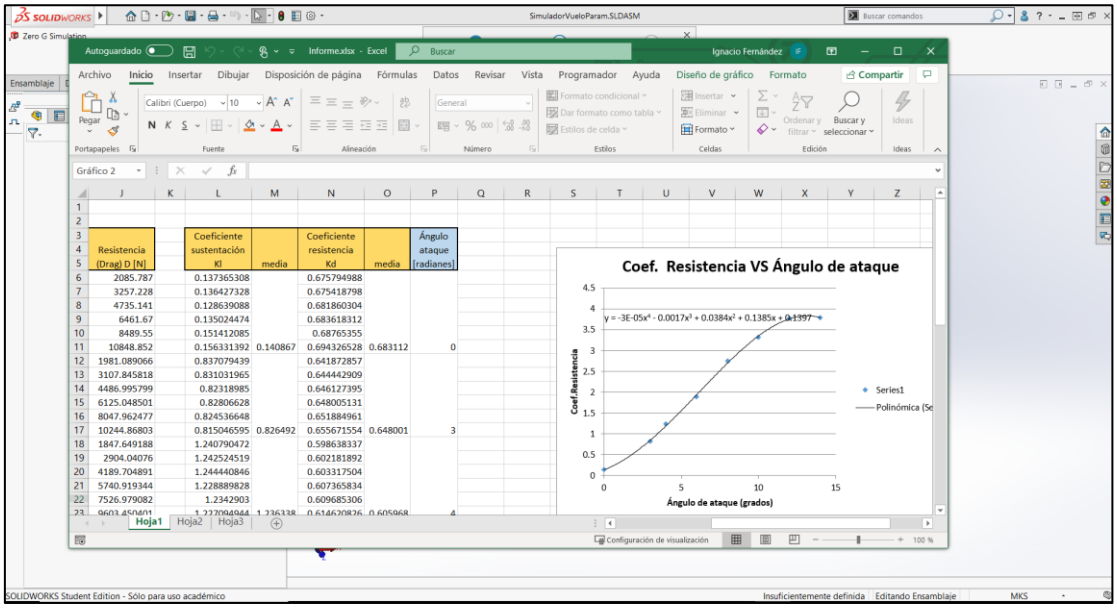


El siguiente paso es rellenar todos los campos del formulario.

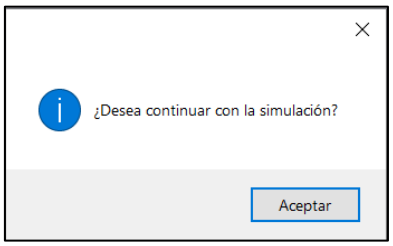
1. Introduzca los ángulos de ataque (en radianes) que se han utilizado en el estudio de *Flow Simulation.*
2. Introduzca las velocidades (en km/h) que se han utilizado en el estudio de *Flow Simulation.*
3. Cargue el archivo (formato .csv) con los resultados de las fuerzas en el eje Y y Z provenientes del estudio de *Flow Simulation*.

4. Cargue el archivo (formato .csv) con los impulsos (fuerza de empuje) que recibe la aeronave que desea estudiar, durante los 16 segundos de vuelo parabólico.
5. Introduzca la velocidad inicial (en m/s) durante los 0,2 segundos iniciales de vuelo parabólico
6. Cargue el archivo (formato .csv) con los ángulos de ataque a los que se ve sometida la aeronave que desea estudiar, durante los 16 segundos de vuelo parabólico.
7. Introduzca el peso (en kg) de la aeronave con combustible antes de despegar.
8. Clique el botón "CALCULAR" para proceder a la simulación.

En primer lugar, se abrirá una ventana de Excel y después se abrirá SOLIDWORKS® *Motion Simulation*.



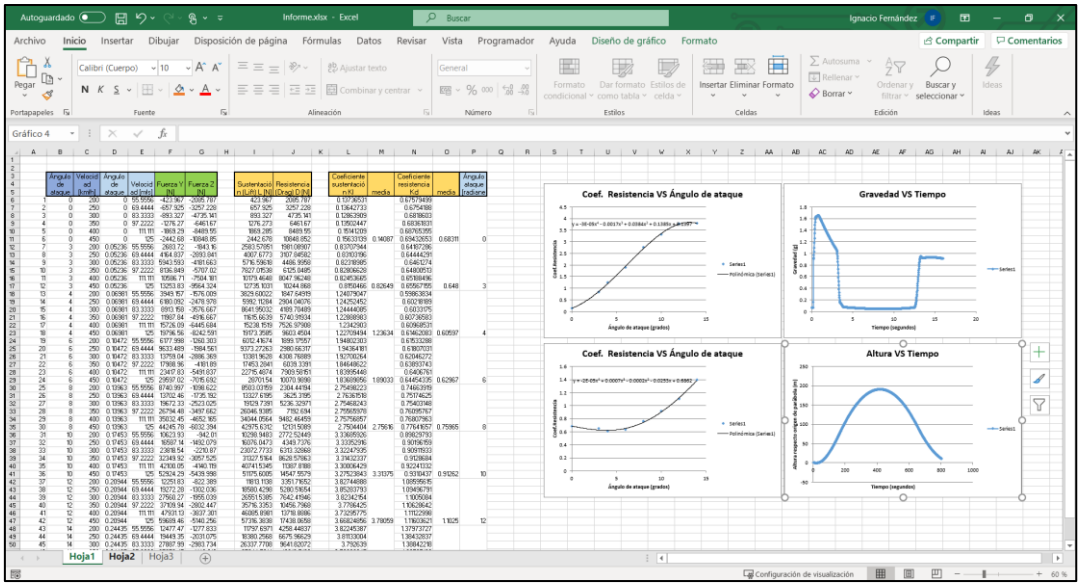
Deje trabajar al programa hasta que le aparezca un mensaje como el mostrado en la (Figura x)

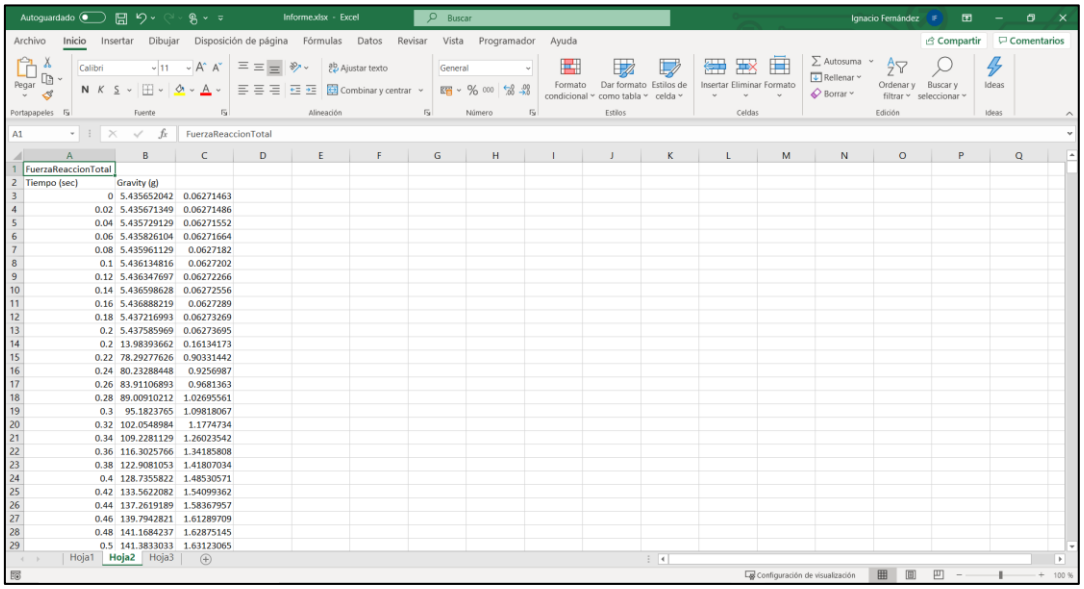


Cuando clique **Aceptar**, se ejecutará la simulación, que tarda escasos minutos.

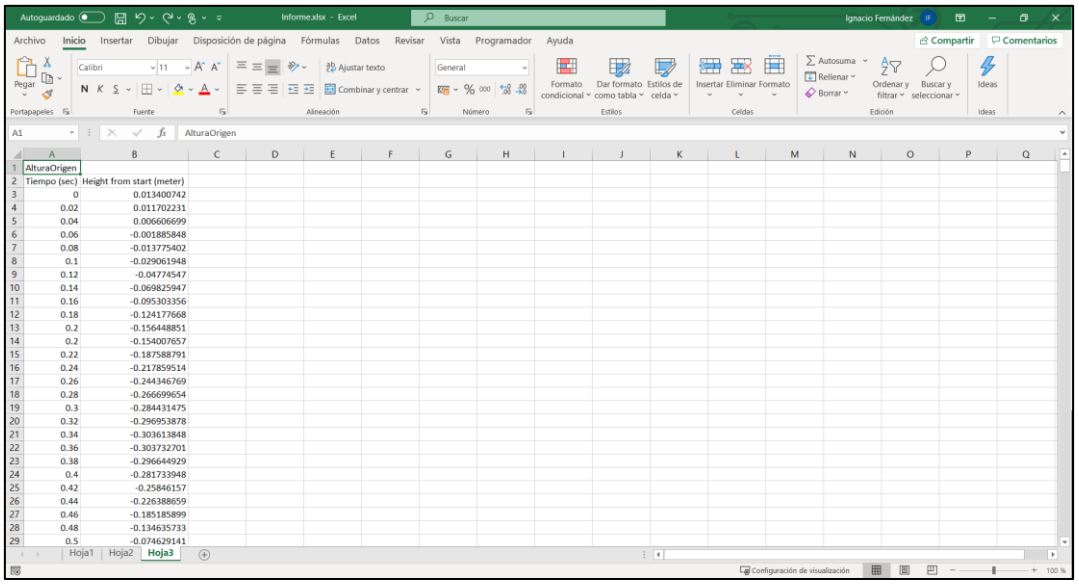
Deje que finalice por completo y abra el Libro de Excel, donde tendrá el informe con los datos y gráficos computados (Figura x). En las hojas 2 (Figura x) y 3 (Figura x) estarán los resultados de FuerzaReaccionTotal y los de AlturaOrigen respectivamente.

Si lo desea, puede guardar el documento Excel.





FuerzaReaccionTotal		
Tiempo (sec)	Gravity (g)	
0	5.435652042	0.06271463
0.02	5.435671349	0.06271486
0.04	5.435729129	0.06271552
0.06	5.435826104	0.06271664
0.08	5.435961129	0.0627182
0.1	5.436134816	0.0627202
0.12	5.436347697	0.06272266
0.14	5.436598628	0.06272556
0.16	5.436888219	0.0627289
0.18	5.437216993	0.06273269
0.2	5.437585969	0.06273695
0.2	13.98393662	0.16134173
0.22	78.29277626	0.90331442
0.24	80.23288448	0.9256987
0.26	83.91106893	0.9681363
0.28	89.00910212	1.02695561
0.3	95.1823765	1.09818067
0.32	102.0548984	1.1774734
0.34	109.2281129	1.26023542
0.36	116.3025766	1.34185808
0.38	122.9081053	1.41807034
0.4	128.7355822	1.48530571
0.42	133.5622082	1.54099362
0.44	137.2619189	1.58367957
0.46	139.7942821	1.61289709
0.48	141.1684237	1.62875145
0.5	141.3833033	1.63123065



AlturaOrigen	
Tiempo (sec)	Height from start (meter)
0	0.013400742
0.02	0.011702231
0.04	0.006606699
0.06	-0.001885848
0.08	-0.013775402
0.1	-0.029061948
0.12	-0.04774547
0.14	-0.069825947
0.16	-0.095303356
0.18	-0.124177668
0.2	-0.156448851
0.2	-0.154007657
0.22	-0.187588791
0.24	-0.217859514
0.26	-0.244346769
0.28	-0.266699654
0.3	-0.284431475
0.32	-0.296953878
0.34	-0.303613848
0.36	-0.303732701
0.38	-0.296644929
0.4	-0.281733948
0.42	-0.25846157
0.44	-0.226388659
0.46	-0.185185899
0.48	-0.134635733
0.5	-0.074629141

El ensamblaje, con su simulación, se cerrará automáticamente sin guardar para evitar que el archivo se dañe. Por favor, espere a que eso ocurra.