



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE CIENCIAS

SIMULACIÓN NUMÉRICA DE LA DINÁMICA DE UN  
COHETE HÍBRIDO

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

MATEMÁTICA

P R E S E N T A :

NATALIA EDITH MEJÍA BAUTISTA

TUTOR

DRA. URSULA X. ITURRARÁN VIVEROS

COTUTOR

DR. JUAN CLAUDIO TOLEDO ROY

CIUDAD UNIVERSITARIA, CDMX, JUNIO 2025





*«No creas lo que tus ojos te dicen. Todo lo que muestran son limitaciones. Mira con tu comprensión, encuentra lo que ya sabes y verás el camino para volar.»*

Juan Salvador Gaviota



# Agradecimientos

Este trabajo no habría sido posible sin el apoyo, la compañía y la inspiración de muchas personas que, de distintas formas, me ayudaron a llegar hasta aquí.

Agradezco profundamente a mis padres, por su amor incondicional, por enseñarme el valor del esfuerzo y por sostenerme incluso en los momentos en que dudaba de mí. A mi hermano Carlos, gracias por cada gesto de cuidado, por llevarme comida cuando me quedaba trabajando largas horas, y por recordarme que no estoy sola.

A mis amigas Camila y Paola, por su cariño, por escucharme y por celebrar conmigo cada pequeño avance.

Agradezco a todos los profesores que me han guiado y creído en mí incluso cuando yo no lo hacía, a la Dra. Úrsula Xiomara Iturrarán Viveros, por su acompañamiento, su paciencia y su confianza en este proceso.

A la Dirección General de Asuntos del Personal Académico (DGAPA), y en particular al proyecto PAPIIT IN111823, por el apoyo económico que hizo posible este trabajo.

Y muy especialmente, al Dr. Claudio Toledo, por compartir conmigo la pasión por la cohetería experimental. Gracias por las conversaciones que desataron ideas, por ayudarme a seguir desarrollando el software, y por soñar juntos con llevar la simulación desde un Jupyter notebook hasta un proyecto real.

Gracias a todos por caminar conmigo hasta este punto. Este trabajo también es de ustedes.



# Índice general

Agradecimientos	v
Índice general	vii
Introducción	xI
Objetivos	xIII
<b>1. Antecedentes</b>	<b>1</b>
1.1. Modelación y simulación . . . . .	1
1.1.1. Simulación en el ciclo de diseño . . . . .	1
1.1.2. Estructura de los Modelos de Simulación . . . . .	3
1.2. Vehículos lanzadores o cohetes . . . . .	5
1.2.1. Cohetes tipo sonda . . . . .	5
1.2.2. Aplicaciones de los cohetes tipo sonda . . . . .	6
1.2.3. Estructura general . . . . .	6
1.2.4. Fases de vuelo . . . . .	7
1.3. Contexto histórico de la cohetería . . . . .	9
1.3.1. Mecánica clásica y balística . . . . .	9
1.3.2. Sector aeroespacial en México . . . . .	10
1.3.3. Seguridad y regulación . . . . .	13
1.4. Equipo Propulsión UNAM . . . . .	14
1.5. Simuladores de vuelo . . . . .	15
1.5.1. Software existente . . . . .	17
<b>2. Modelado dinámico</b>	<b>23</b>
2.1. Conceptos fundamentales sobre ecuaciones diferenciales ordinarias . .	23
2.1.1. Ecuaciones diferenciales ordinarias y problemas de valores iniciales . . . . .	23
2.1.2. Problemas rígidos . . . . .	24
2.1.3. Importancia del modelado con EDO . . . . .	24
2.2. Ecuaciones diferenciales en dinámica . . . . .	25
2.3. Modelos dinámicos simples . . . . .	25
2.3.1. CASO 1: Tiro con gravedad . . . . .	25

2.3.2. CASO 2: Tiro parabólico con gravedad y arrastre lineal . . . . .	27
2.3.3. CASO 3: Tiro parabólico con gravedad y arrastre cuadrático . . . . .	30
2.4. Formulación del modelo completo . . . . .	35
2.4.1. Dinámica de vuelo . . . . .	36
2.4.2. Fuerzas involucradas . . . . .	38
2.4.3. Consideraciones angulares . . . . .	44
2.4.4. Ecuaciones . . . . .	47
<b>3. Métodos numéricos</b>	<b>51</b>
3.1. Interpolación lineal . . . . .	52
3.1.1. Aplicación a tablas de datos . . . . .	52
3.1.2. Implementación computacional . . . . .	53
3.2. Solución numérica de ecuaciones diferenciales ordinarias . . . . .	54
3.2.1. Métodos no adaptativos . . . . .	54
3.2.2. Métodos adaptativos explícitos . . . . .	56
3.2.3. Métodos adaptativos implícitos y automáticos . . . . .	58
3.3. Fuentes de error . . . . .	60
3.4. Control de error . . . . .	60
3.4.1. Estimación de error local y global . . . . .	60
3.4.2. Control de paso . . . . .	61
3.5. Propiedades . . . . .	62
3.5.1. Precisión y orden de los métodos . . . . .	62
3.5.2. Estabilidad numérica . . . . .	63
3.5.3. Convergencia . . . . .	64
3.6. Eficiencia . . . . .	64
3.7. Análisis teórico . . . . .	65
3.7.1. Criterios de comparacion . . . . .	65
3.8. Validacion . . . . .	66
3.8.1. Condiciones iniciales y parámetros de simulación . . . . .	66
3.8.2. Validez de la comparacion numerica entre los 40 casos evaluados	66
3.8.3. Caso 1: Tiro vertical con gravedad . . . . .	68
3.8.4. Caso 2: Tiro parabólico con arrastre lineal . . . . .	70
3.8.5. Caso 3: Tiro parabólico con arrastre cuadrático . . . . .	73
3.9. Discusión técnica y recomendaciones . . . . .	76
3.10. Elección de integrador . . . . .	77
<b>4. Simulador de trayectoria</b>	<b>79</b>
4.1. Suposiciones . . . . .	80
4.2. Diseño del simulador . . . . .	80
4.3. Desarrollo del simulador . . . . .	83
4.3.1. Clases utilizadas y sus métodos . . . . .	84
4.4. Implementación del simulador . . . . .	88
4.5. Interfaz gráfica . . . . .	89
4.5.1. Aplicaciones educativas y uso en equipos estudiantiles . . . . .	91
4.6. Análisis estático y dinámico: seguridad del vuelo . . . . .	92

4.6.1. Análisis estático . . . . .	92
4.6.2. Análisis dinámico . . . . .	92
4.6.3. Relación con la seguridad del lanzamiento . . . . .	92
4.7. Limitaciones de la simulación . . . . .	93
<b>5. Caso Xitle 2</b>	<b>95</b>
5.1. Caso práctico: Xitle 2 . . . . .	95
5.1.1. Datos del motor . . . . .	97
5.1.2. Adquisición de datos . . . . .	97
5.2. Modelado del vehículo . . . . .	100
5.3. Vuelo Experimental . . . . .	100
5.3.1. Datos del GPS . . . . .	101
5.4. Vuelo Openrocket . . . . .	102
5.5. Vuelo simulado . . . . .	104
5.6. Verificación . . . . .	109
5.6.1. Comparación de resultados . . . . .	109
5.7. Validación . . . . .	112
5.7.1. Prueba 1. Vuelo CON paracaídas . . . . .	112
5.7.2. Prueba 2. Vehículo inestable . . . . .	113
5.7.3. Prueba 3. Simulación con y sin viento . . . . .	114
5.7.4. Prueba 4. Variación de parámetros de diseño . . . . .	117
<b>6. Conclusiones</b>	<b>119</b>
<b>Bibliografía</b>	<b>121</b>
<b>Índice de figuras</b>	<b>125</b>
<b>Índice de tablas</b>	<b>127</b>
<b>Apéndice A. Física de la dinámica de vuelo</b>	<b>129</b>
A.0.1. Sistemas de coordenadas . . . . .	129
A.0.2. Cinemática . . . . .	132
A.1. Modelo atmosférico . . . . .	135
<b>Apéndice B. Método de Dormand-Prince 8(5,3)</b>	<b>137</b>
B.1. Esquema general . . . . .	137
B.2. Tabla de Butcher . . . . .	137
B.3. Pesos y estimaciones embebidas . . . . .	138
<b>Apéndice C. Repositorio de código en Github</b>	<b>141</b>
C.1. Instrucciones de uso del simulador . . . . .	141
C.1.1. Requisitos del sistema . . . . .	141
C.1.2. Instalación de dependencias . . . . .	141
C.1.3. Ejecución del simulador . . . . .	141
C.1.4. Estructura general del repositorio . . . . .	142

C.1.5. Licencia y colaboración . . . . .	142
--	-----

# Introducción

Actualmente, el sector aeroespacial comienza a consolidarse en México, tomando cada vez mayor relevancia en el ámbito científico y tecnológico. En esta búsqueda por el desarrollo y la innovación, es indispensable asegurar la viabilidad y seguridad del diseño de lanzadores espaciales, especialmente en entornos experimentales donde los recursos son limitados y cada prueba representa un esfuerzo considerable.

Dentro de este panorama, la modelación matemática y la simulación numérica emergen como herramientas fundamentales. Los simuladores de trayectoria permiten anticipar el comportamiento dinámico de un vehículo lanzado al espacio, resolviendo sistemas de ecuaciones diferenciales que describen su evolución temporal bajo distintas condiciones físicas. Este enfoque no solo posibilita evaluar con precisión la respuesta del sistema ante perturbaciones, como ráfagas de viento, sino también refinar el diseño a través de un proceso iterativo fundamentado en cálculos rigurosos.

Simular trayectorias con modelos físicos bien planteados y métodos numéricos adecuados permite generar predicciones confiables sin la necesidad de realizar múltiples lanzamientos reales. Esto reduce significativamente el tiempo y los recursos requeridos, al mismo tiempo que incrementa la seguridad operativa. Además, contribuye a identificar zonas de riesgo, estimar áreas de impacto y definir protocolos de evacuación con base en resultados cuantitativos.

La motivación de este trabajo es comprender a fondo el sustento físico y matemático detrás del desarrollo de simuladores de trayectoria o corredores de vuelo comerciales, con el objetivo de construir una herramienta propia, confiable y ajustable a las necesidades de proyectos reales. Se busca no solo replicar trayectorias plausibles, sino también garantizar que el modelo numérico refleje con fidelidad la dinámica de vuelo y permita tomar decisiones informadas durante el proceso de diseño.

Esta tesis se propone, por tanto, establecer una base teórica robusta de mecánica de vuelo y de integración numérica, describiendo los modelos empleados para caracterizar el movimiento de cohetes suborbitales. Se implementan distintos métodos de resolución de EDOs, analizando sus ventajas, limitaciones y desempeño en escenarios concretos. El objetivo es lograr simulaciones suficientemente precisas que nos guíen hacia un diseño óptimo, cumpliendo con los requerimientos de misión y garantizando vuelos estables incluso ante condiciones adversas.

Aunado a esto, el equipo estudiantil “Propulsión UNAM” trabaja actualmente en el desarrollo, manufactura y pruebas de cohetes híbridos como *Xitle* y *Xitle II*, diseñados para alcanzar alturas superiores a los 9 km y velocidades supersónicas. Contar con un simulador propio, adaptable y matemáticamente sólido es esencial para acompañar este proceso, fortalecer las capacidades técnicas y asegurar que cada iteración en el diseño esté respaldada por fundamentos rigurosos.

# Objetivos

- Modelar el vuelo de cohetes suborbitales mediante ecuaciones diferenciales.
- Diseñar e implementar un simulador numérico usando un integrador adecuado.
- Analizar el efecto de diferentes integradores y parámetros en las soluciones obtenidas.
- Determinar el integrador óptimo para las necesidades de simulación.
- Validar los resultados obtenidos.
- Obtener predicciones detalladas de la trayectoria y orientación del vehículo.
- Brindar una herramienta fácil de adaptar y usar para el diseño de nuevos vehículos.
- Brindar mayor seguridad al planear lanzamientos futuros al predecir posibles zonas de impacto.



# 1 Antecedentes

En este capítulo se aborda la necesidad de realizar modelos y simulaciones computacionales en los procesos de desarrollo de productos de ingeniería. Así, como los pasos a seguir para obtener un resultado apropiado.

Por otro lado, se introducen los conceptos básicos relacionados a los cohetes y la importancia particular de simular sus trayectorias.

Finalmente, se hacen presentes algunos antecedentes nacionales del sector aeroespacial y softwares comerciales que realizan análisis similares.

## 1.1. Modelación y simulación

En el mundo actual nos vemos envueltos en una gran cantidad de procesos, fenómenos y situaciones complejas. Por esta razón, puede dificultarse entender en dónde estamos y cómo podemos lograr nuestros objetivos. Para esto, es necesario conocer cómo funcionan internamente los sistemas, de modo que podamos afectar el elemento indicado y obtener el resultado que esperamos, tal como indica Shannon en [Shannon, 1988].

### 1.1.1. Simulación en el ciclo de diseño

A lo largo del ciclo de diseño una etapa con numerosos beneficios es la simulación de los sistemas o productos.

Cuando se lleva a cabo de manera efectiva, esta metodología puede ser de gran ayuda para mejorar y perfeccionar nuestros sistemas. No obstante, una ejecución deficiente puede conducir a interpretaciones erróneas y conclusiones precipitadas.

Por consiguiente, se convierte en una herramienta sumamente poderosa para respaldar la toma de decisiones. Su capacidad para modelar sistemas complejos y proporcionar información detallada de cada proceso en un entorno virtual controlado amplía aún más su utilidad. Sin embargo, no siempre es posible variar los parámetros que requerimos si nuestro modelo no es lo suficientemente robusto o detallado.

#### Ventajas

Diseño asistido por computadora con simulación en la Ingeniería

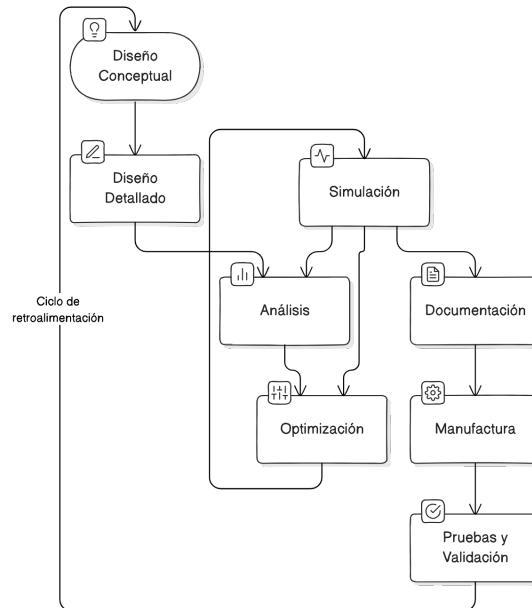


Figura 1.1: Ciclo de diseño asistido por simulación en la Ingeniería  
Realizado con IA en Eraser.io

- Permite descomponer a los sistemas en componentes más simples y estudiar su comportamiento de manera aislada antes de integrarlos en un modelo completo.
- Optimización del diseño: Permite identificar y corregir problemas antes de la manufactura.
- Mejora la seguridad: Permite realizar pruebas de forma más económica y segura, evitando los riesgos de hacer las pruebas en un entorno real.
- Reducción u optimización de costos, tiempo y recursos: Permite anticipar y evaluar posibles cambios en nuestro sistema real.
- Aceleración de desarrollo: Permite probar nuevas tecnologías y conceptos de forma rápida y eficiente.

## Desventajas

- Requiere datos precisos y confiables para alimentar el modelo. Los resultados están sujetos a la calidad de los datos y la precisión del modelo.
- El desarrollo y la ejecución de modelos de simulación pueden demandar tiempo y recursos significativos.
- Difícil explorar todas las alternativas o todas las variantes que pueden existir dentro del sistema.

### 1.1.2. Estructura de los Modelos de Simulación

Los modelos de simulación constan de componentes esenciales: entradas, procesos y salidas. Las entradas representan variables que afectan al sistema, como datos de entrada o condiciones iniciales. Los procesos describen cómo se comporta el sistema, y las salidas son los resultados que se obtienen de la simulación. Comprender esta estructura es fundamental para diseñar y analizar modelos de manera efectiva.

Para capturar la esencia del sistema sin sobrecargar el modelo con detalles innecesarios es necesario tomar en cuenta las siguientes etapas:

1. Planteamiento del problema y definir los objetivos. ¿Qué escenarios se simularán? ¿Qué información deseamos obtener?
2. Investigar y recopilar datos del sistema.
3. Identificar las entidades, sus características y relaciones más relevantes en el sistema.
4. Simplificar el modelo conceptual y selección del modelo. Tomar algunas suposiciones que no afecten considerablemente el fenómeno que se va a estudiar.
5. Selección de variables.
6. Derivar ecuaciones y algoritmos que describan cómo estas variables interactúan entre sí.
7. Codificación de un software apropiado.
8. Implementación del software creado.
9. Validar el modelo y verificar los resultados obtenidos.
10. Análisis y síntesis.

De la observación se deben establecer condiciones mínimas en las cuales el sistema opera teóricamente y con esto se propone un punto de partida para el proceso de investigación.

La simulación implica la evaluación de los resultados de la simulación para obtener información interpretable físicamente sobre el sistema.

Recordemos que un modelo matemático puede ser tan complejo como uno quiera, o en su caso, requiera. Por ende, se debe tener en cuenta el error que va a tener y si las hipótesis son apropiadas al fenómeno a estudiar.

Para esto, es importante identificar correctamente las variables de estados y los parámetros del sistema.

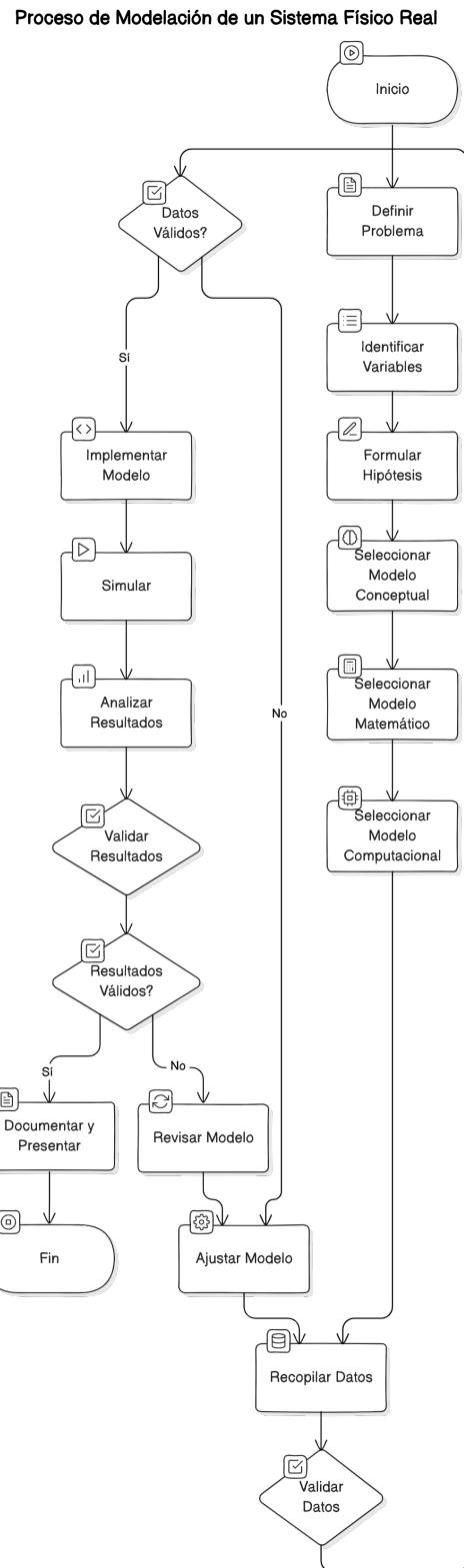


Figura 1.2: Proceso de Modelación

## 1.2. Vehículos lanzadores o cohetes

La siguiente sección introduce al lector al contexto del desarrollo aeroespacial en México, así como a los conceptos básicos, funcionamiento general y aplicaciones de los cohetes suborbitales, particularmente de los cohetes tipo sonda.

De acuerdo a la NOM-009-SCT2 2009: Especificaciones especiales y de compatibilidad para el almacenamiento y transporte de las sustancias, materiales y residuos peligrosos de la clase 1 explosivos de la Secretaría de Comunicaciones y Transportes [de Comunicaciones y Transportes, 2010], un cohete se define de la siguiente manera:

**“COHETES.-** Objetos constituidos por un motor de cohete y una carga útil, que puede ser una cabeza de combate explosiva u otro dispositivo. Este término comprende los proyectiles dirigidos y COHETES con cabeza inerte; COHETES con carga explosiva; COHETES con carga impulsora; COHETES DE COMBUSTIBLE LÍQUIDO con carga explosiva; COHETES LANZACABOS.”

Por otro lado, de acuerdo a la Guía de Procedimientos y Operaciones Estandarizadas ENMICE 2023 [de ENMICE, 2023], un **vehículo lanzador** está definido como un conjunto de sistemas (piezas, equipos y componentes) necesarios para integrar de manera funcional un medio de transporte propulsado por un motor tipo cohete que permite el traslado de personas o cosas entre la superficie de la Tierra y un lugar en la atmósfera o más allá en el espacio.

### 1.2.1. Cohetes tipo sonda

Un cohete de sondeo o tipo sonda se define de acuerdo a la Federal Aviation Administration’s Office of Commercial Space Transportation (FAA/AST) como “Un cohete pequeño diseñado para hacer misiones de investigación, como tomar medidas y realizar experimentos científicos durante un vuelo sub-orbital, en beneficio comercial, gubernamental, militar o de clientes privados”.

A diferencia de un vehículo de lanzamiento, no entra en órbita terrestre, sino que realiza una trayectoria balística describiendo un vuelo sub-orbital, similar a una parábola. Además, difiere de los cohetes amateur porque alcanzan altitudes mayores.

Los cohetes sonda son utilizados para transportar instrumentos desde los 10-50 kilómetros (estratosfera) hasta los 1500 km sobre la superficie de la tierra (exosfera); generalmente la altitud entre los globos meteorológicos y los satélites, regiones importantes del espacio que son “demasiado bajas” para ser muestradas por satélites.

A los instrumentos de investigación dentro del cohete se le conoce como “*payload*” o, en español, “Carga útil” y esta característica los hace relevantes, pues permite adaptarse a múltiples objetivos sin la necesidad de hacer grandes inversiones monetarias, pues los costos son significativamente menores a las misiones de orbitadores.

### 1.2.2. Aplicaciones de los cohetes tipo sonda

La importancia de los experimentos realizados en los cohetes como vehículos radica en dos características:

- Su capacidad de alcanzar grandes velocidades y aceleraciones.
- Su capacidad de funcionar en el vacío.

Lo anterior permite una apertura a una variedad de sectores, como son salud, calidad ambiental, telecomunicaciones, seguridad y defensa, entre otros, lo cual constituye un avance tecnológico beneficioso para los países.

Algunas aplicaciones específicas son:

1. Meteorología: Estudio más preciso de parámetros físicos, químicos y biológicos de la atmósfera alta cuyo acceso era antes complicado.
2. Epidemiología: Estudio de perfiles de gases, temperatura, humedad, radiación y presión para el conocimiento de las condiciones ambientales y ciclos de reproducción de microorganismos patógenos.
3. Medicina aeroespacial: Estudio de los efectos de la hipergravedad sobre el sistema nervioso central, lo cual contribuye a la investigación de la fisiología en vuelos tripulados de aviones supersónicos o cohetes espaciales.
4. Agricultura: Control biológico de cultivos y asentamientos ilegales de marihuana, diseminación de bacterias y hongos o de fertilizantes en campos de semillas.
5. Monitoreo de contaminación: Muestreo y análisis de la troposfera y del aire para prevenir problemas de contaminación.
6. Mejoramiento de las telecomunicaciones: Apoyar la calibración de antenas de transmisión de radio, TV y telefonía móvil.
7. Colocación en posiciones sub orbitales y orbitales de satélites o cubesats: Posicionamiento de picosatélites modulares de órbita baja y circular para observación de la Tierra, monitoreo ambiental, rastreo del espacio aéreo y ultraterrestre suprayacente con fines de seguridad y defensa.

### 1.2.3. Estructura general

Un cohete tipo sonda se puede representar como una estructura integrada por 7 componentes principales o sistemas, los cuales corresponden a:

1. Ojiva (nariz): Componente frontal de forma aerodinámica cuya función es reducir el arrastre.
2. Fuselaje: Componente estructural, generalmente tubular, dentro del cual se resguardan otros componentes.

3. Sistema de recuperación (paracaídas): Mecanismos y dispositivos diseñados para asegurar la integridad del vehículo mediante una caída controlada.
4. Instrumentos (computadora de vuelo, antenas, cámaras): Sistemas encargados de monitorear y medir los datos durante el vuelo.
5. Carga útil: Instrumento científico o experimento controlado que se lleva acabo durante la misión.
6. Motor cohete (cámara de combustión, propelente y tobera): Sistema que genera la fuerza necesaria para impulsar el vehículo.

De acuerdo a su configuración y tipo de combustible, pueden clasificarse como sólidos, líquidos o híbridos de una o varias etapas.

- Sólido
- Híbrido
- Líquido

7. Sistemas de control de vuelo (aletas, boattail): componentes que sirven como superficies de control aerodinámico

En efecto, la Figura 1.3 ilustra la estructura general de un cohete sonda de motor híbrido. En el caso de los vehículos de motor sólido se omite el tanque de oxidante y la bahía de válvulas.

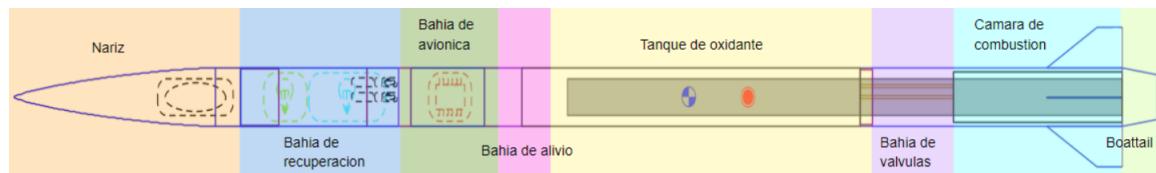


Figura 1.3: Estructura general de un cohete híbrido

#### 1.2.4. Fases de vuelo

Durante el vuelo se presentan distintas fases o etapas determinadas por eventos específicos, que describen el tipo de movimiento en el cual se encuentra el vehículo.

Las principales fases y los eventos que las determinan se enuncian a continuación:

- Evento 0. Colocación y preparación en torre de lanzamiento
- Evento 1. Ignición: Momento en el que se enciende la carga que inicia al motor.
- Evento 2. Despegue: Momento en el que el cohete se levanta de su posición de reposo impulsado por el empuje del motor.

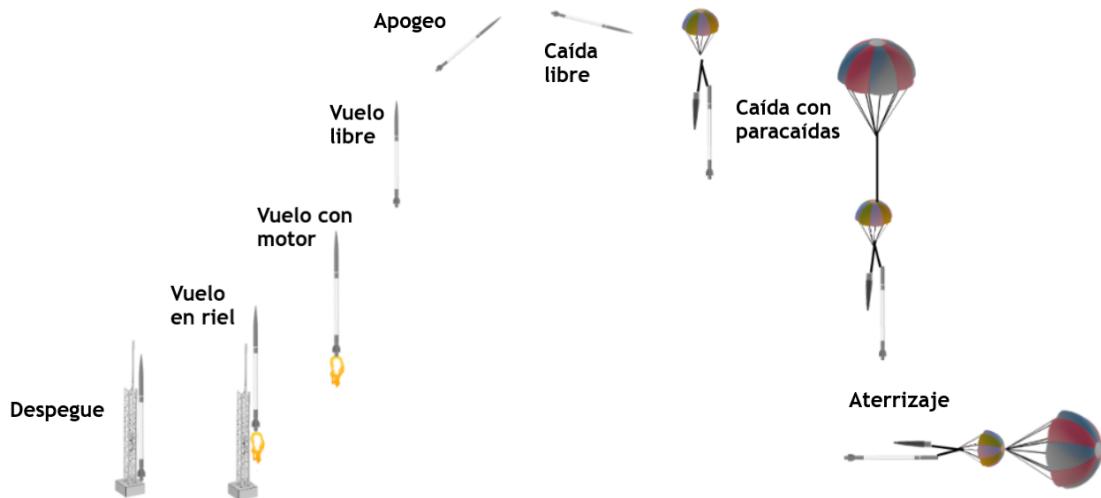


Figura 1.4: Diagrama de fases de vuelo.

Imagen modificada del reporte técnico de Propulsión UNAM

- **FASE 1: Vuelo sobre el riel de lanzamiento** Etapa en la que el cohete se mueve únicamente sobre el riel, restringido debido a que sigue unido a la base de lanzamiento mediante las guías de la estructura.
- Evento 3. Salida del riel
- **FASE 2. Vuelo ascendente con motor prendido** Etapa en la que el motor sigue produciendo empuje y por ende acelerando al vehículo y ya no hay restricción en el movimiento.
- Evento 4. Apagado del motor o *Main Engine Cut Off (MECO)*
- **FASE 3. Vuelo ascendente con motor apagado** Etapa en la que el motor ya está apagado, por ende no hay empuje.
- Evento 5. Llegada al APOGEO.
- Evento 6. Activación del sistema de recuperación (puede ser de una o varias etapas)
- **FASE 4. Recuperación: Vuelo descendente con paracaídas** En esta etapa se despliega el paracaídas de arrastre y posteriormente el principal. Entonces, las ecuaciones de movimiento empleadas en las fases anteriores ya no son adecuadas. Se considera como caída de un punto de masa con un coeficiente de arrastre determinado.
- Evento 7. Llegada al suelo o ATERRIZAJE

## 1.3. Contexto histórico de la cohetería

### 1.3.1. Mecánica clásica y balística

Vale la pena recordar que cualquier simulación balística se remonta a la **física clásica**: Galileo Galilei (1638), Isaac Newton (1687), Benjamin Robins (1742), entre otros. Cada uno aportando parte fundamental de los modelos que se usan hoy en día.

#### Galileo Galilei (1564–1642): Cinemática de la caída libre

En *Discorsi e dimostrazioni matematiche intorno a due nuove scienze* (1638), Galileo encontró que todos los objetos caen con la misma aceleración si se desprecia la resistencia del aire. Además, propuso que el movimiento en dos dimensiones, como el tiro parabólico, puede descomponerse en dos partes independientes:

- un **movimiento horizontal** a velocidad constante  $v_0 \cos \theta$ , y
- un **movimiento vertical** con aceleración constante  $g$  (gravedad).

Al combinar (superponer) ambos movimientos se obtiene la trayectoria *parabólica* completa.

$$y(x) = x \tan \theta - \frac{g x^2}{2v_0^2 \cos^2 \theta},$$

primer modelo de tiro balístico al vacío.

#### Isaac Newton (1642–1727): Ley de gravitación y órbitas

En los *Principia Mathematica* (1687), Newton resolvió el *problema de dos cuerpos*, mostrando que la atracción gravitatoria  $F_g = GMm/r^2$  genera trayectorias que son secciones cónicas y estableciendo el valor de la aceleración local de la gravedad  $g = GM/R^2$  aún usada como referencia.

Cuando Edmund Halley le preguntó cómo conocía la forma de la órbita planetaria, Newton respondió: “*I calculated it.*” (“la calculé”), que es lo que se busca hacer en este trabajo.

#### Benjamin Robins (1707–1751): Resistencia del aire y medición de velocidad

El ingeniero británico Benjamin Robins publicó *New Principles of Gunnery* (1742), donde demostró que la fuerza de arrastre es aproximadamente proporcional a  $v^2$  en el rango subsónico-supersónico, introduciendo la resistencia cuadrática que se emplea en los cálculos y el simulador de esta tesis.

### 1.3.2. Sector aeroespacial en México

#### Felipe Ángeles (1908–1914)

El **General Felipe de Jesús Ángeles Ramírez** (1868–1919) es considerado el padre de la balística científica en México. Como director de la *Escuela de Tiro* del Ejército (1908) impartió mecánica analítica, balística interior y exterior, y publicó su influyente *Teoría del Tiro* [Ángeles Ramírez, 1908]. Entre sus principales aportes destacan:

- Formulación de ecuaciones con fricción lineal ( $-\mu v$ ) y cuadrática ( $-\mu v|v|$ ) para estimar alcances con diferente munición;
- Primeras *tablas de dispersión* mexicanas, antecedente directo del concepto CEP (*circular error probable*);
- Introducción del *tiro indirecto* y métodos de corrección probabilística, con lo que redujo hasta 40 % el gasto de proyectiles en campaña.

Su combinación de rigurosidad matemática y experimentación de campo sirve hoy como *benchmark* analítico para los casos simples simulados en esta tesis (tiro vertical, parabólico con arrastres lineal y cuadrático).

#### Cabo Tuna y los pioneros potosinos (1957–1962)

En la madrugada del **28 de diciembre de 1957**, los doctores **Gustavo del Castillo y Gama y Candelario Pérez Rosales**, recién egresados de Purdue, lanzaron el *Física I* en un paraje de Las Charcas (S.L.P.) apodado con humor “*Cabo Tuna*”. El cohete (1.70 m, 8 kg, propelente azufre–zinc) alcanzó  $\sim 2.5$  km y transportó placas fotográficas para estudiar rayos cósmicos. Entre 1958 y 1961 se construyeron los *Física II* y *III* (hasta 12 km de apogeo), financiados mediante becas estatales y premios de la Sociedad Mexicana de Física [Arauz Lara, 2022].

Estos éxitos motivaron la creación, en 1962, de la **Comisión Nacional del Espacio Exterior** (CNEE).

#### La Comisión Nacional del Espacio Exterior (CNEE, 1962–1977)

Dependiente de la Secretaría de Comunicaciones y Transportes, la CNEE fue promulgada por decreto presidencial el **26 de septiembre de 1962**. Su comité técnico, presidido por **Walter Cross Buchanan** (IPN) y **Porfirio Becerril Buitrón** (IPN), estableció tres líneas de trabajo [de Comunicaciones y Transportes, 1964]:

1. *Programa de Sondeos Atmosféricos*: 27 cohetes de fabricación nacional tipo SCT-1/2 (195 kg, 35 km) para medición de densidad y temperatura troposférica;

2. *Estudio de telecomunicaciones satelitales*: convenio con la NASA para estaciones de rastreo *STADAN* en Guaymas;
3. *Formación de recursos humanos*: becas de instrumentación y propulsión en el Goddard Space Flight Center.

Aunque la CNEE cesó operaciones en 1977, sentó precedentes de política espacial y generó la primera *Red Nacional de Estaciones Estratosféricas*.

### Sociedad Universitaria de Cohetes Experimentales (SUCE, 1969–1975)

Fundada por estudiantes de la UNAM e IPN, la **SUCE** integró a químicos, biólogos e ingenieros con un objetivo común: desarrollar vehículos de sondeo atmosférico a bajo costo.

- Construyeron seis prototipos —*Azteca*, *Jaguar*, *Océlotl*— con alcances entre 3 y 10 km.
- Introdujeron cámaras de 16 mm y sensores de ozono, pioneras en medición ambiental universitaria [Sociedad Universitaria de Cohetes Experimentales, 1973].
- Publicaron el *Manual de Propulsión de Pólvoras Compuestas* (1972), que circuló en escuelas de ingeniería por más de dos décadas.

El grupo se disolvió tras un accidente en 1975, pero su legado inspiró a equipos actuales como *Propulsión UNAM*.

### UNAM 1973: Diseño de un cohete de propelente líquido

La tesis colectiva *Diseño y construcción de un cohete de propelente líquido para sondeo atmosférico* [Muñoz et al., 1973] fue elaborada por **Carlos R. Duarte Muñoz**, **Marco A. Flores Verdugo**, **Héctor C. Hernández García**, **Ronald V. M. López Gómez**, **Augusto Sánchez Cifuentes** y **José de J. Vega Jiménez**, bajo la dirección del Ing. Dagoberto de la Serna Valdivia en la Facultad de Ingeniería de la UNAM (1973). Fue un *proyecto universitario mexicano* que propuso un vehículo sonda de **propelente líquido (LOX–JP-4)** y se planteó como respuesta académica a la pausa en los programas de la CNEE.

### El Seminario de Dinámica y Sistemas Complejos del Dr. Alberto Alonso y Coria (1974–1980)

El físico-matemático **Alberto Alonso y Coria** (FC-UNAM) fundó en 1974 el *Seminario de Dinámica y Sistemas Complejos*, un espacio pionero donde se introdujeron herramientas numéricas para resolver ecuaciones diferenciales en problemas balísticos y orbitales [Alonso y Coria, 1977].

- *Códigos analógico–digitales*: implementación de los métodos de Euler, Runge–Kutta de orden 4 y extrapolación de Bulirsch–Stoer en la computadora analógica EAI-TR10, resolviendo trayectorias con arrastre lineal y cuadrático.
- *Materiales para propulsión*: construcción del primer espectrómetro Mössbauer de la Facultad de Ciencias para caracterizar aceros refractarios en toberas de cohetes [Alonso y Coria and García, 1975].
- *Cursos intersemestrales “Dinámica de Cohetes”*: apuntes DCR-1/DCR-2 (1976–1980) que cubren balística exterior, parámetros orbitales e integración adaptativa; antecedente directo de los actuales cursos de astrodinámica en la FC-UNAM.
- *Vinculación con la SUCE*: los algoritmos desarrollados se compartieron con la Sociedad Universitaria de Cohetes Experimentales para estimar apogeos de los prototipos *Azteca* y *Jaguar*.

El legado del Dr. Alonso y Coria persiste en la formación de varias generaciones de físicos e ingenieros, mostrando el ciclo de docencia–investigación–aplicación que inspira este trabajo.

### **La Agencia Espacial Mexicana (AEM, 2010–presente)**

La **Agencia Espacial Mexicana** fue creada mediante la *Ley que crea la AEM* publicada el 30 de julio de 2010 en el Diario Oficial de la Federación [AEM, 2010]. Como organismo descentralizado de la Secretaría de Infraestructura, Comunicaciones y Transportes, que articula las políticas nacionales en torno a cinco ejes del *Programa Nacional de Actividades Espaciales* (PNAE):

1. Formación de capital humano
2. Investigación y desarrollo
3. Competitividad industrial
4. Seguridad y asuntos internacionales
5. Financiamiento–gestión de la información.

De esta forma, la AEM ha tenido proyectos tangibles que combinan participación universitaria, alianzas internacionales y desarrollo industrial. A continuación se resumen algunas de las iniciativas, mostrando que el sector aeroespacial toma cada vez más fuerza en el país.

- **AztechSat-1** (*UPAEP–NASA*, 2019): primer CubeSat mexicano desplegado desde la EEI; validó un enlace satélite–satélite con la constelación GlobalStar [Smith et al., 2021].
- **Firma de los Acuerdos Artemisa** (2021): integra a México en la coalición para el regreso humano a la Luna y misiones a Marte [de Relaciones Exteriores, 2021].

- **Misión lunar COLMENA** (UNAM–AEM, 2022–2024): cinco microrrobots de 60 g se lanzaron a bordo del lander *Peregrine* (Astrobotic) el 8 ene 2024; una fuga de propelente impidió el alunizaje y la nave reentró en la atmósfera terrestre el 18 ene 2024, por lo que la primera exploración robótica mexicana sobre regolito lunar quedó pendiente. [Duarte et al., 2022].
- **GXIBA-1** (UPAEP–AEM–JAXA, 2024): segundo CubeSat nacional; medirá aerosoles volcánicos desde la EEI a partir de 2025 [UPAEP, 2024].
- **Impulso a clústeres aeroespaciales** (Querétaro, Baja California, Bajío): atracción de IED y convenios academia–industria para manufactura, ensayos de propulsión y simulación digital avanzada [Mexicana, 2025].

### 1.3.3. Seguridad y regulación

El lanzamiento de un cohete, incluso de carácter universitario, tiene riesgos potenciales para la población, la infraestructura y el tráfico aéreo. Por esto, existe normativa, internacional y nacional, que exige presentar *modelos de dispersión, análisis probabilísticos de riesgo y corredores de exclusión* antes de autorizar cualquier operación. A continuación se menciona el **marco jurídico** más relevante y se explica cómo los simuladores numéricos, como el desarrollado en esta tesis, contribuyen a lograr dichos requisitos.

#### Normativa internacional de referencia

La norma **14 CFR Parte 420** de la Federal Aviation Administration (FAA), Code of Federal Regulations, Title 14, *License to Operate a Launch Site*. Regula la concesión de licencias para operar plataformas de lanzamiento en territorio estadounidense y se ha convertido en el estándar para la seguridad pública en vuelos espaciales comerciales y académicos, estableciendo lo siguiente:

1. Índice de víctimas esperadas  $E_c = \sum_i P_i C_i < 10^{-4}$ ;
2. Definición de *Launch Safety Zones* y *flight corridors* que minimicen el riesgo a terceros;
3. Análisis con software de dispersión validado y simulaciones Monte-Carlo;
4. Requiere un *Flight Safety Analysis Report* que documente supuestos, métodos y verificación del código.

#### Marco legal mexicano (en consolidación)

Las regulaciones nacionales, actualmente en elaboración conjunta AEM-AFAC— se inspiran directamente en sus análogas estadounidenses.

- *Ley de Aviación Civil* y su Reglamento (reforma 2022): artículos 56–59 bis introducen el concepto de *vehículo aeroespacial reutilizable* y asignan a la **AFAC** la expedición de permisos.
- *Proyecto NMX-AEM-NX-01* (consulta pública 2025): adopta el mismo umbral de seguridad que la Parte 420.
- *Acuerdo AEM-AFAC 01/2023*: establece el **Permiso Temporal de Operaciones Espaciales (PTOE)** y su expediente en seis apartados:
  1. *Manifestación de Impacto Ambiental (MIA)* ante SEMARNAT.
  2. Análisis de *trayectoria nominal y dispersión anómala* mediante simulación Monte-Carlo (criterio  $Ec < 10^{-4}$ ).
  3. Contratación de *seguro de responsabilidad civil* conforme al valor de la carga y el riesgo poblacional.
  4. Publicación de avisos **NOTAM/NOTMAR**.
  5. Presentación de *plan de contingencia y rescate*.
  6. Registro del objeto espacial en el *Registro Nacional de Objetos Lanzados al Espacio Ultraterrestre*.

Para sustentar los futuros lanzamientos, la AEM desarrolló en 2024 un **Corredor de Vuelo**, que propone zonas de exclusión sobre cartografía nacional. Este sistema funciona ya como referencia técnica para los permisos temporales de operaciones espaciales (PTOE) y es base en el análisis de futuras zonas de lanzamiento en territorio mexicano.

Así, la tesis no solo aporta rigor científico, sino también *cumplimiento regulatorio*, reforzando la seguridad de futuros lanzamientos en territorio mexicano.

## 1.4. Equipo Propulsión UNAM

El equipo **Propulsión UNAM** es una agrupación estudiantil de la Asociación Aeroespacial de la Facultad de Ingeniería, fundado en **agosto de 2020**, en plena pandemia, por ocho estudiantes —*incluida la autora de esta tesis*— con el propósito de crear capital humano e infraestructura para el diseño, manufactura y prueba de cohetes experimentales de combustible sólido e híbrido.

Siendo así, la **división de cohetería experimental de alta potencia de la Universidad Nacional Autónoma de México**. Los estudiantes se involucran en el ciclo completo de diseño, desarrollo, pruebas y manufactura, donde mejoran sus capacidades sobre las áreas técnicas como: propulsión, estructuras, aerodinámica y electrónica embebida; así como sus habilidades blandas (liderazgo, planeación de proyectos, logística) y de manufactura.

El equipo ha obtenido **reconocimiento nacional e internacional** y tiene como meta ser la primera organización estudiantil latinoamericana en llegar al espacio, con un cohete capaz de cruzar la línea de Karman. A la fecha (2025), se han lanzado más de 10 **cohetes Komodo** de combustión sólida con apogeo entre 500 m y 1 km. También, se finalizó el **Proyecto Insurgente** de combustible sólido con apogeo de 3 km y 4 kg de carga útil y el **proyecto Xitle y Xitle II** con apogeo estimado de 9 km y 4 kg de carga útil.

En la Figura 1.5, se puede observar a las tres primeras generaciones del equipo Propulsión UNAM sosteniendo un *cohete Komodo*, el *Proyecto Insurgente* y el *Proyecto Xitle*, de abajo hacia arriba.



Figura 1.5: Equipo Propulsión UNAM y sus cohetes. Temporada 2023

## 1.5. Simuladores de vuelo

**Def:** La **trayectoria** de un vehículo suborbital es la secuencia de posiciones espaciales, medidas con relación a la superficie terrestre, que sigue durante su vuelo.

Un **simulador de vuelo** es un programa de cómputo que simula el comportamiento y la dinámica de vuelo de cohetes sub-orbitales. Estos simuladores se utilizan para probar y validar el diseño y rendimiento de nuevas configuraciones de vehículos antes de ser construidos y puestos en vuelo.

Brindan una opción segura y poco costosa de evaluar la estabilidad y estructura general de los vehículos bajo múltiples condiciones de vuelo.

Algunos rasgos principales de los simuladores incluyen:

- Modelación Física y aerodinámica lo más realista
- Configuración de vehículos personalizable: Formas y dimensiones variables. Consideración a distintos tipos de propelente.
- Diversos escenarios de vuelo: Múltiples condiciones atmosféricas, de viento, de sitio de lanzamiento, inclinación variable del riel de lanzamiento, entre otras.
- Análisis de datos y visualización de resultados: Gráficas, tablas y animaciones de los resultados obtenidos.

Para describir la orientación del cohete en el espacio tridimensional se emplean los siguientes ejes o momentos de giro:

- Alabeo o *roll*: Movimiento en el que el cohete gira sobre su eje longitudinal
- *pitch* o cabeceo: Movimiento en el que el cohete inclina su punta/nariz hacia la izquierda o derecha
- Guiñada o *yaw*: Movimiento en el que el cohete inclina su punta/nariz hacia delante o hacia atrás.

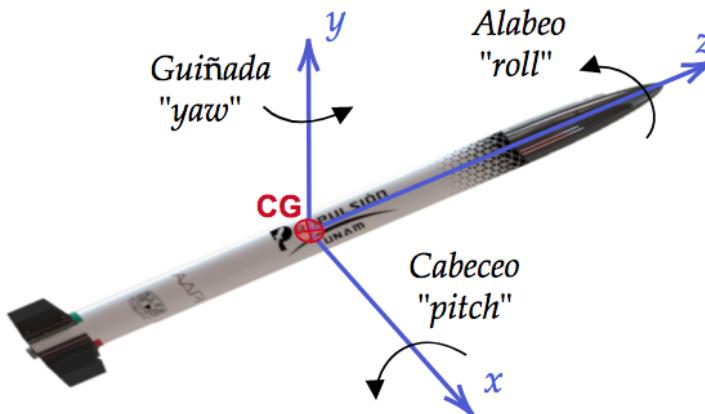


Figura 1.6: Momentos principales: Cabeceo, guiñada y alabeo

Los simuladores de vuelo se clasifican de acuerdo a la complejidad del movimiento que se modela. Más específicamente, el número de variables que se deben resolver para describir el movimiento de un sistema conocido como **grados de libertad (GDL)** o *Degrees Of Freedom (DOF)*:

- **1GDL**: Ofrece aproximaciones en una sola dirección, en este caso hacia arriba y hacia abajo, del movimiento del vehículo. Generalmente, considera al vehículo como un punto de masa y es útil para aproximaciones iniciales y conocer orden de magnitud de los principales parámetros o fuerzas involucrados.

Un ejemplo bastante completo es presentado en el Capítulo 3 de [Rubio, 2023] .

- **3GDL:** Ofrece aproximaciones de movimiento dentro del plano 2D (arriba/abajo y derecha/izquierda) y modela también el ángulo de orientación del vehículo en torno a uno de sus ejes de rotación. En estos simuladores, los vehículos pueden considerarse como punto de masa o como cuerpos extendidos.

Este tipo de simuladores ignoran fenómenos como el giro sobre el eje longitudinal del vehículo o las trayectorias e inclinaciones que se pudieran presentar fuera del plano.

- **6GDL:** Este tipo de simulador ofrece movimiento completo en todas las direcciones espaciales, translación y rotación en los tres ejes, por lo que brinda una aproximación de vuelo más realista pero computacionalmente más complejas.

### 1.5.1. Software existente

Existen *softwares* comerciales desarrollados por grupos de individuos aficionados a la cohetería o instituciones académicas. A pesar de que algunos son de código abierto y de alta confiabilidad, no es posible saber cómo se realizan algunos de los cálculos y no permiten realizar modificaciones con facilidad. Lo cual no permite al usuario modificar libremente los parámetros de acuerdo a sus intereses de análisis.

Con cierta práctica y lectura de la documentación adecuada se pueden modificar ciertos parámetros. Sin embargo, se considera que realizar un simulador propio fomenta una mejor comprensión de los fenómenos involucrados y se estará más familiarizado con la estructura del código para modificaciones o agregar funciones posteriores

Los principales simuladores usados en la comunidad de la cohetería experimental son los siguientes.

#### OpenRocket

[Niskanen, 2009] , [Niskanen, 2013] Simulador gratuito y de código abierto más popular entre la comunidad de cohetería. Es fácil de usar por su interfaz amigable y permite simular una amplia gama de cohetes al permitir agregar componentes de forma libre. Utiliza Runge-Kutta de 4to orden para integrar las ecuaciones de movimiento e incluye modelos realistas de viento, así como varias opciones para las aproximaciones de la Tierra y sus condiciones atmosféricas.

#### RocketPy

[Ceotto Giovani H., 2021] Código abierto en Python que permite simular en 6 grados de libertad. Es una buena opción para usuarios que tienen experiencia en programación y actualmente está en mejora constante por la comunidad en *Github*.

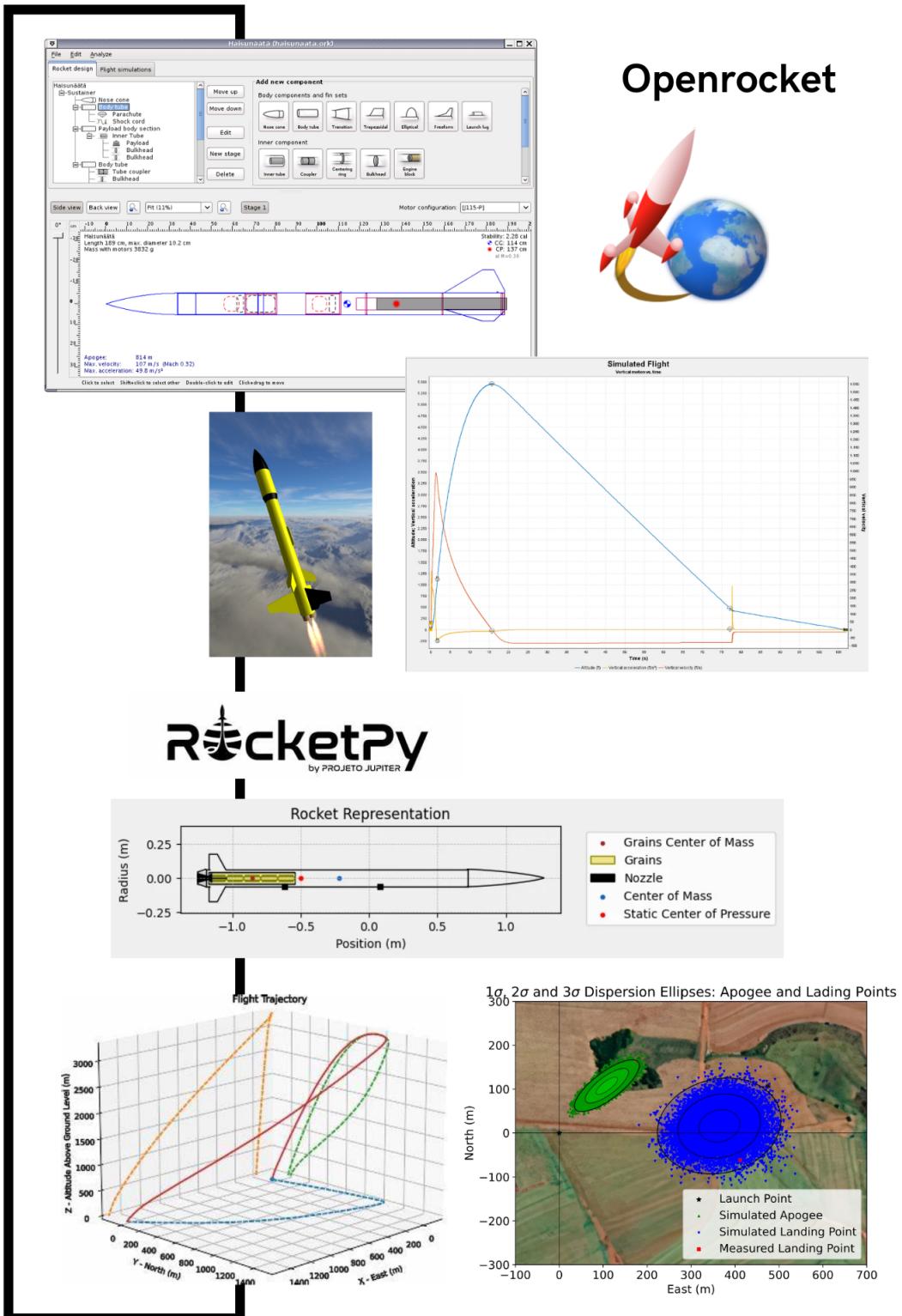


Figura 1.7: Principales softwares comerciales

## RockSim y RockSimPRO

[Tim, 1997] Simuladores comerciales desarrollados por **Apogee Components** que permiten usar componentes de figuras personalizadas y hacer análisis de estabilidad dinámica al calcular la variación de masa del motor. La versión de paga es *RockSim-PRO*, la cual permite una visualización en un mapa y análisis de dispersión. Además permite una visualización en vídeo de la animación de la trayectoria de vuelo.

## RASAero

[Charles E. Rogers, 2019] RASAero es un software de análisis aerodinámico que permite predecir con precisión y flexibilidad los coeficientes aerodinámicos necesarios para simular las trayectorias. Incluye una amplia base de datos de componentes y parámetros basados en experimentos de túnel de viento, entre otros.

## Cambridge Rocketry Simulator

[Simon Box, 2009], [Simon Box, 2010] Simulador comercial que ofrece versatilidad para usarse en Matlab. Permite realizar simulaciones estocásticas y optimización del ángulo de lanzamiento para obtener un mejor apogeo.

## Recapitulación

Todos los softwares mencionados cuentan con diferentes características técnicas de simulación y varían los métodos para modelar los fenómenos más importantes de los vuelos. En la tabla ?? se resumen el lenguaje de programación y los grados de libertad de cada uno de los softwares mencionados.

Simulador / Características	Lenguaje	GDL
OpenRocket	Java	6
RockSim	C++ (Qt)	3
RockSimPRO	C++ (Qt)	6
RocketPy	Python	6
CambridgeRocketrySimulator	Java GUI / Python	6
RASAero II	C# (.NET 3.5)	3
AeroVector	Python	3

Tabla 1.1: Comparativa de simuladores comerciales

Sumado a lo anterior, el equipo Propulsión UNAM ha desarrollado con anterioridad simuladores de vuelo de 1 grado de libertad implementados en Matlab o Excel, como expone Aztatzi Rubio Isaac en [Rubio, 2023] que permitió conocer los principales parámetros de vuelo para los cohetes Komodo o Insurgente.

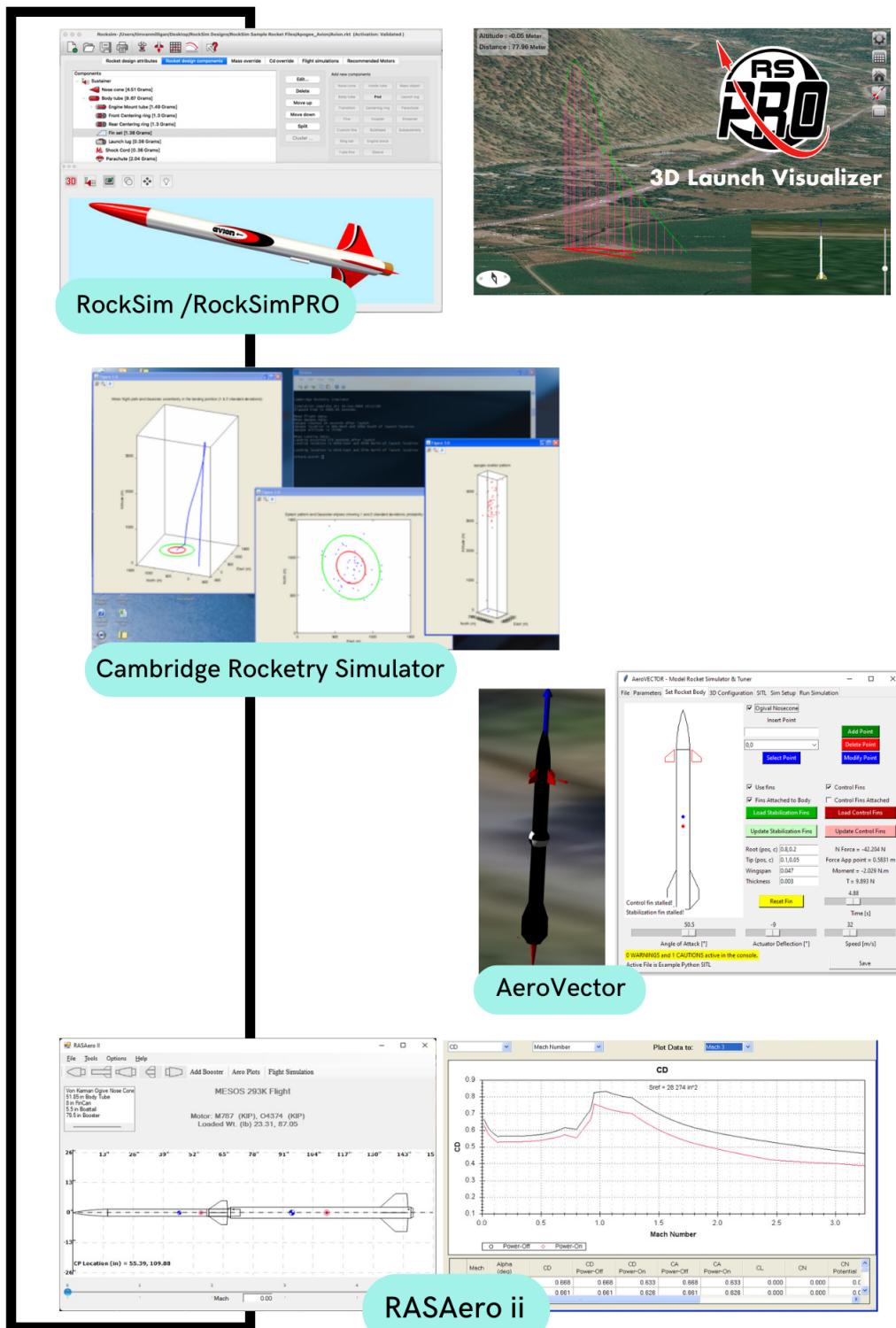


Figura 1.8: Otros softwares comerciales

El simulador desarrollado en este trabajo es considerado de 3GDL pero agregando un análisis trigonométrico apropiado puede extenderse a un simulador de 6GDL.



## 2 Modelado Dinámico mediante Ecuaciones Diferenciales Ordinarias

El estudio de la dinámica de vuelo de un cohete requiere la formulación y resolución de un sistema de **ecuaciones diferenciales ordinarias** (EDO) que describen la evolución temporal de las variables de estado del sistema, tales como la posición, velocidad, orientación y velocidad angular del vehículo. Estas ecuaciones se derivan del balance de fuerzas y momentos que actúan sobre el cohete y constituyen la base para la predicción y análisis de su comportamiento dinámico.

Con el objetivo de introducir progresivamente los conceptos y validar la metodología, este capítulo presenta primero casos simplificados y su respectiva solución analítica, para después abordar la formulación completa del modelo dinámico del cohete.

El desarrollo progresivo de estos modelos permite, en los capítulos posteriores, la implementación y comparación de diversos métodos numéricos para la resolución eficiente y precisa de los sistemas de EDO que rigen la dinámica de vuelo.

### 2.1. Conceptos fundamentales sobre ecuaciones diferenciales ordinarias

El modelado de la dinámica de vuelo mediante ecuaciones diferenciales ordinarias requiere un conocimiento básico de los elementos matemáticos involucrados, así como de las dificultades que pueden presentarse al resolver estos sistemas.

#### 2.1.1. Ecuaciones diferenciales ordinarias y problemas de valores iniciales

Una ecuación diferencial ordinaria (EDO) es una ecuación que relaciona una función desconocida con sus derivadas respecto a una única variable independiente. En el caso de sistemas dinámicos, esta variable es el tiempo  $t$ .

El estudio de la dinámica de un sistema físico frecuentemente conduce a un **problema de valores iniciales** (PVI), también denominado **problema de Cauchy**, cuya formulación general es:

$$\frac{d\vec{y}}{dt} = \vec{f}(t, \vec{y}), \quad \vec{y}(t_0) = \vec{y}_0, \quad (2.1)$$

donde  $\vec{y}(t) \in \mathbb{R}^n$  representa el vector de variables de estado del sistema, y  $\vec{f}$  define la dinámica en función del tiempo y del estado actual.

La resolución de un PVI consiste en encontrar la función  $\vec{y}(t)$  que satisface la ecuación diferencial y cumple la condición inicial  $\vec{y}(t_0) = \vec{y}_0$ . [Teschl, 2012]

### 2.1.2. Problemas rígidos

En algunos casos, el sistema dinámico presenta un comportamiento denominado **rigidez**. Un sistema de EDO's se considera rígido cuando existen escalas de tiempo muy diferentes que obligan al uso de pasos de integración extremadamente pequeños para garantizar la estabilidad numérica [Hairer and Wanner, 1996], aun cuando la solución evoluciona lentamente.

En el contexto de la dinámica de vuelo, la rigidez puede originarse por:

- La diferencia de órdenes de magnitud entre las aceleraciones lineales y las aceleraciones angulares.
- La inclusión de términos disipativos, como el arrastre cuadrático o el momento amortiguador, que actúan en escalas de tiempo pequeñas.
- La presencia de discontinuidades o cambios bruscos en las condiciones de vuelo, como el despliegue del paracaídas.

Este tipo de problemas requiere la utilización de **métodos numéricos adecuados**, tales como esquemas implícitos o integradores adaptativos, que permitan resolver las ecuaciones con estabilidad y eficiencia.

### 2.1.3. Importancia del modelado con EDO

La correcta formulación y resolución de los sistemas de EDO permite predecir con precisión la trayectoria y orientación del cohete, así como analizar la influencia de distintas condiciones iniciales y parámetros físicos. En este trabajo, los casos simplificados presentados a continuación sirven para ilustrar los conceptos fundamentales y validar los métodos numéricos empleados, antes de abordar la simulación del modelo dinámico completo.

## 2.2. Ecuaciones diferenciales en dinámica

Las ecuaciones diferenciales ordinarias permiten que se describa la dependencia en el tiempo de múltiples fenómenos, en nuestro caso el movimiento dado por el vuelo del vehículo.

Para establecer un modelo de ecuaciones diferenciales es necesario expresar el estado  $x(t)$  en términos de las derivadas y posteriormente establecer las condiciones iniciales apropiadas.

Se parte de la segunda Ley de Newton que conecta la aceleración  $a(t)$  del cuerpo con la fuerza  $F$  que actúa sobre una masa  $m$ :

$$a(t) = \frac{F}{m} \quad (2.2)$$

Recordemos que la aceleración es la segunda derivada de la posición y que la fuerza puede depender del tiempo, de la posición y de la velocidad del cuerpo, entonces  $x''(t) = a(t)$  y  $F(t, x, x')$ .

Con lo anterior, podemos reescribir la ecuación [2.2] como:

$$x''(t) = \frac{F(t, x, x')}{m} \quad (2.3)$$

## 2.3. Modelos dinámicos simples

A continuación se muestran los casos simplificados de la dinámica de una masa puntual en una o dos direcciones considerando algunas fuerzas. Cabe destacar que cada uno de estos casos cuenta con solución analítica gracias a las simplificaciones de desacoplamiento realizadas.

### 2.3.1. CASO 1: Tiro con gravedad

El caso más sencillo es el que considera una fuerza constante. Sea  $m$  la masa de la partícula,  $g$  la aceleración de la gravedad y  $y(t)$  la altura de la partícula en función del tiempo  $t$ . La ecuación diferencial ordinaria que describe el movimiento es:

$$\frac{d^2y(t)}{dt^2} = -g \quad (2.4)$$

Como vimos anteriormente, para convertir esta EDO a un sistema de EDOs de primer orden, definimos una nueva variable  $v$ , que corresponde a la velocidad, de la siguiente manera:  $v = \frac{dy}{dt}$

Luego, podemos escribir la EDO original [2.4] como:

$$\frac{dv}{dt} = -g$$

Resultando en un sistema de dos ecuaciones diferenciales ordinarias de primer orden equivalente:

$$\begin{cases} \frac{dy}{dt} = v, \\ \frac{dv}{dt} = -g. \end{cases} \quad (2.5)$$

### Solución analítica

Para obtener la solución analítica de este caso se procede con los siguientes pasos. Integrando la ecuación con respecto a  $t$  para obtener la velocidad:

$$\int \frac{d^2y}{dt^2} dt = \int -g dt \quad (2.6)$$

$$\Rightarrow v(t) = \frac{dy}{dt} = -gt + C_1 \quad (2.7)$$

Luego, al integrar la ecuación con respecto a  $t$  para obtener la posición:

$$\int \frac{dy}{dt} dt = \int (-gt + C_1) dt \quad (2.8)$$

$$\Rightarrow y(t) = -\frac{1}{2}gt^2 + C_1t + C_2, \quad (2.9)$$

donde  $C_1$  y  $C_2$  son constantes de integración que en nuestro contexto están asociadas a las condiciones iniciales de velocidad  $v_0$  y posición  $y_0$ . Obteniendo,

$$v(t) = -gt + v_0 \quad (2.10)$$

$$y(t) = -\frac{1}{2}gt^2 + v_0t + y_0 \quad (2.11)$$

### Apogeo y tiempo de apogeo

Para encontrar el valor máximo de la posición, es necesario determinar el valor de  $t$  que maximiza la función  $y(t)$ . Para ello, tomamos la ecuación de la velocidad y la igualamos a cero:

$$\frac{dy}{dt} = v_0 - gt = 0 \Rightarrow v_0 = gt \quad (2.12)$$

Resolviendo para  $t$ , obtenemos el instante en que se alcanza la altura máxima.

$$t = \frac{v_0}{g} \quad (2.13)$$

Ahora, podemos sustituir este valor de  $t$  en la ecuación [2.9] de  $y(t)$  para encontrar la altura máxima:

$$y_{max} = y\left(\frac{v_0}{g}\right) = y_0 + v_0 \left(\frac{v_0}{g}\right) - \frac{1}{2}g \left(\frac{v_0}{g}\right)^2 \quad (2.14)$$

$$= y_0 + \frac{v_0^2}{g} - \frac{v_0^2}{2g} = y_0 + \frac{2v_0^2 - v_0^2}{2g} \quad (2.15)$$

$$\Rightarrow y_{max} = y_0 + \frac{v_0^2}{2g}, \quad (2.16)$$

donde  $y_{max}$  es la altura máxima que alcanza el objeto.

### 2.3.2. CASO 2: Tiro parabólico con gravedad y arrastre lineal

Considerando ahora otra fuerza relevante, como lo es el arrastre. Sea  $m$  la masa de la partícula,  $k$  el coeficiente de arrastre y  $v(t)$  la velocidad de la partícula en función del tiempo  $t$ . La ecuación diferencial ordinaria que describe el movimiento en dos dimensiones es:

En el eje  $x$ :

$$\frac{dv_x}{dt} = -\frac{k}{m}v_x \quad (2.17)$$

En el eje  $y$ :

$$\frac{dv_y}{dt} = -g - \frac{k}{m}v_y \quad (2.18)$$

Observemos que la fuerza de arrastre automáticamente ajusta su dirección dependiendo del movimiento:

- **Cuando el objeto sube ( $v_y > 0$ ):** el arrastre actúa hacia abajo, reduciendo  $v_y$ .
- **Cuando el objeto cae ( $v_y < 0$ ):** el arrastre actúa hacia arriba, oponiéndose al movimiento

Por lo tanto, el signo del arrastre no necesita cambiar manualmente, ya que depende del signo de  $v_y$ .

#### Soluciones analíticas

En esta sección se integran analíticamente las ecuaciones para obtener los valores de las velocidades y posiciones en cada tiempo  $t$ .

**Solución para  $v_x(t)$** 

La ecuación diferencial para  $v_x(t)$  se puede escribir como:

$$\frac{dv_x}{dt} + \frac{k}{m}v_x = 0 \quad (2.19)$$

Resolviendo por separación de variables:

$$\frac{1}{v_x}dv_x = -\frac{k}{m}dt \quad (2.20)$$

Integrando ambos lados:

$$\ln |v_x| = -\frac{k}{m}t + C_1 \quad (2.21)$$

Exponenciando:

$$v_x(t) = C_2 e^{-\frac{k}{m}t} \quad (2.22)$$

Usando la condición inicial  $v_x(0) = v_{x0}$  se sigue que  $v_x(0) = C_2 e^0 = C_2$ . De donde,

$$v_x(t) = v_{x0} e^{-\frac{k}{m}t} \quad (2.23)$$

**Solución para  $v_y(t)$** 

La ecuación diferencial para  $v_y(t)$  también se puede escribir como:

$$\frac{dv_y}{dt} + \frac{k}{m}v_y = -g \quad (2.24)$$

Usamos el factor integrante:

$$\mu(t) = e^{\int \frac{k}{m} dt} = e^{\frac{k}{m}t} \quad (2.25)$$

Multiplicando por el factor integrante:

$$e^{\frac{k}{m}t} \frac{dv_y}{dt} + e^{\frac{k}{m}t} \frac{k}{m}v_y = -ge^{\frac{k}{m}t} \quad (2.26)$$

El lado izquierdo es una derivada:

$$\frac{d}{dt} \left( v_y e^{\frac{k}{m}t} \right) = -ge^{\frac{k}{m}t} \quad (2.27)$$

Integrando ambos lados:

$$v_y e^{\frac{k}{m}t} = -\frac{gm}{k} e^{\frac{k}{m}t} + C_3 \quad (2.28)$$

Dividiendo por  $e^{\frac{k}{m}t}$ :

$$v_y(t) = -\frac{gm}{k} + C_3 e^{-\frac{k}{m}t} \quad (2.29)$$

Usando la condición inicial  $v_y(0) = v_{y0}$ , calculamos  $C_3$ :

$$v_y(0) = -\frac{gm}{k} + C_3 e^0 = -\frac{gm}{k} + C_3 \quad (2.30)$$

$$\Rightarrow C_3 = v_{y0} + \frac{gm}{k} \quad (2.31)$$

Por lo tanto:

$$v_y(t) = \left( v_{y0} + \frac{gm}{k} \right) e^{-\frac{k}{m}t} - \frac{gm}{k} \quad (2.32)$$

### Solución para $x(t)$

Recordemos que la relación entre  $x(t)$  y  $v_x(t)$  es:

$$\frac{dx}{dt} = v_x(t) \quad (2.33)$$

Sustituyendo  $v_x(t)$  de la ecuación 2.23:

$$\frac{dx}{dt} = v_{x0} e^{-\frac{k}{m}t} \quad (2.34)$$

Integrando ambos lados:

$$x(t) = \int v_{x0} e^{-\frac{k}{m}t} dt \quad (2.35)$$

Resolviendo la integral:

$$x(t) = -\frac{v_{x0}m}{k} e^{-\frac{k}{m}t} + C_4 \quad (2.36)$$

Usando la condición inicial  $x(0) = 0$ , se sigue que

$$x(0) = \frac{-v_{x0}m}{k} e^0 + C_4 = \frac{-v_{x0}m}{k} + C_4 \quad (2.37)$$

$$\Rightarrow C_4 = \frac{v_{x0}m}{k} \quad (2.38)$$

Por lo tanto:

$$x(t) = \frac{v_{x0}m}{k} \left( 1 - e^{-\frac{k}{m}t} \right) \quad (2.39)$$

### Solución para $y(t)$

Por otro lado, la relación entre  $y(t)$  y  $v_y(t)$  es:

$$\frac{dy}{dt} = v_y(t) \quad (2.40)$$

Sustituyendo  $v_y(t)$ :

$$\frac{dy}{dt} = \left( v_{y0} + \frac{gm}{k} \right) e^{-\frac{k}{m}t} - \frac{gm}{k} \quad (2.41)$$

Integrandos ambos lados:

$$y(t) = \int \left[ \left( v_{y0} + \frac{gm}{k} \right) e^{-\frac{k}{m}t} - \frac{gm}{k} \right] dt = \int \left( v_{y0} + \frac{gm}{k} \right) e^{-\frac{k}{m}t} dt - \int \frac{gm}{k} dt \quad (2.42)$$

Resolviendo cada término:

$$y(t) = -\frac{m}{k} \left( v_{y0} + \frac{gm}{k} \right) e^{-\frac{k}{m}t} - \frac{gm}{k} t + C_5 \quad (2.43)$$

Usando la condición inicial  $y(0) = 0$ , calculamos  $C_5$ .

$$y(0) = -\frac{m}{k} \left( v_{y0} + \frac{gm}{k} \right) e^0 - \frac{gm}{k} 0 + C_5 \quad (2.44)$$

$$= -\frac{m}{k} \left( v_{y0} + \frac{gm}{k} \right) + C_5 \quad (2.45)$$

$$\Rightarrow C_5 = \frac{m}{k} \left( v_{y0} + \frac{gm}{k} \right) \quad (2.46)$$

Por lo tanto:

$$y(t) = -\frac{m}{k} \left( v_{y0} + \frac{gm}{k} \right) e^{-\frac{k}{m}t} - \frac{gm}{k} t + \frac{m}{k} \left( v_{y0} + \frac{gm}{k} \right) \quad (2.47)$$

$$= \frac{m}{k} \left( v_{y0} + \frac{gm}{k} \right) \left( 1 - e^{-\frac{k}{m}t} \right) - \frac{gm}{k} t = \frac{m}{k} \left[ \left( v_{y0} + \frac{gm}{k} \right) \left( 1 - e^{-\frac{k}{m}t} \right) - gt \right] \quad (2.48)$$

Observemos como en este caso, se introducen términos exponenciales que modifican significativamente las trayectorias, en comparación con el caso sin arrastre. La velocidad y posición en los ejes  $x$  y  $y$  disminuyen de manera exponencial con el tiempo debido al término de arrastre lineal, haciendo que las partículas alcancen distancias y alturas menores. Esto muestra la importancia de considerar fuerzas resistivas.

### 2.3.3. CASO 3: Tiro parabólico con gravedad y arrastre cuadrático

Sabemos que el arrastre no es lineal, así que consideraremos una versión "menos simplista" para notar el aumento en la complejidad de resolver las ecuaciones asociadas.

Como en el caso anterior, el tiro parabólico con gravedad y arrastre cuadrático está gobernado por dos fuerzas principales de acuerdo a la componente. De esa forma, se obtienen las siguientes ecuaciones de movimiento:

Separando en componentes  $x$  y  $y$ , con  $\vec{v} = (v_x, v_y)$ , tenemos:

**Eje x:**

$$\frac{dv_x}{dt} = -\frac{k'}{m} |\vec{v}| v_x = -k |\vec{v}| v_x, \quad (2.49)$$

**Eje y:**

$$\frac{dv_y}{dt} = -g - \frac{k'}{m} |\vec{v}| v_y = -g - k |\vec{v}| v_y, \quad (2.50)$$

donde

$$|\vec{v}| = \sqrt{v_x^2 + v_y^2}.$$

$$\text{y } k = \frac{k'}{m}$$

Así, el sistema completo que describe el movimiento es:

$$\frac{dx}{dt} = v_x, \quad (2.51)$$

$$\frac{dy}{dt} = v_y, \quad (2.52)$$

$$\frac{dv_x}{dt} = -k \sqrt{v_x^2 + v_y^2} v_x, \quad (2.53)$$

$$\frac{dv_y}{dt} = -g - k \sqrt{v_x^2 + v_y^2} v_y. \quad (2.54)$$

En general, estas ecuaciones deben resolverse numéricamente debido a la presencia del término no lineal.

### Soluciones analíticas

El sistema con arrastre cuadrático en ambos ejes es no lineal, pero bajo ciertas simplificaciones, podemos obtener expresiones analíticas para  $v_x(t)$ ,  $v_y(t)$ ,  $x(t)$ , y  $y(t)$ .

#### Ecuación en el eje x

La ecuación diferencial para  $v_x(t)$  es:

$$\frac{dv_x}{dt} = -kv_x|v_x|. \quad (2.55)$$

Además, el movimiento en  $x$  es independiente de  $y$ , así que reescribimos como:

$$\frac{1}{v_x|v_x|} dv_x = -k dt. \quad (2.56)$$

Dado que  $v_x|v_x| = v_x^2$  para  $v_x > 0$ , reordenamos:

$$\frac{1}{v_x^2} dv_x = -k dt. \quad (2.57)$$

Integrando ambos lados:

$$\int \frac{1}{v_x^2} dv_x = \int -k dt. \quad (2.58)$$

La integral del lado izquierdo es:

$$-\frac{1}{v_x} = -kt + C_1. \quad (2.59)$$

Resolviendo para  $v_x$ :

$$v_x(t) = \frac{1}{kt - C_1}. \quad (2.60)$$

Usando la condición inicial  $v_x(0) = v_{x0} = \frac{1}{0-C_1} = \frac{1}{-C_1}$ .

Se obtiene  $C_1 = -\frac{1}{v_{x0}}$ . Entonces, multiplicando por  $1 = \frac{v_{x0}}{v_{x0}}$  tenemos:

$$v_x(t) = \frac{1}{kt + \frac{1}{v_{x0}}} \left( \frac{v_{x0}}{v_{x0}} \right) \quad (2.61)$$

$$v_x(t) = \frac{v_{x0}}{1 + kv_{x0}t}. \quad (2.62)$$

Para la posición  $x(t)$ , integramos:

$$x(t) = \int v_x(t) dt = \int \frac{v_{x0}}{1 + kv_{x0}t} dt = v_{x0} \int \frac{1}{1 + kv_{x0}t} dt \quad (2.63)$$

Haciendo el cambio de variable  $\mu = 1 + kv_{x0}t \Rightarrow d\mu = kv_{x0}dt$ . Entonces, la integral nos queda

$$x(t) = v_{x0} \int \frac{1}{\mu} \frac{d\mu}{kv_{x0}} = \frac{v_{x0}}{kv_{x0}} \int \frac{1}{\mu} d\mu = \frac{1}{k} \ln \mu + C \quad (2.64)$$

$$\Rightarrow x(t) = \frac{1}{k} \ln(1 + kv_{x0}t) + C \quad (2.65)$$

Luego, usando la condición inicial  $x(0) = x_0 = \frac{1}{k} \ln 1 + C = \frac{1}{k} 0 + C = C$  obtenemos

$$\Rightarrow x(t) = \frac{1}{k} \ln(1 + kv_{x0}t) + x_0 \quad (2.66)$$

### Ecuación en el eje $y$

La ecuación diferencial para  $v_y(t)$  es:

$$\frac{dv_y}{dt} = -g - kv_y |v_y|. \quad (2.67)$$

Reescribimos como:

$$\frac{1}{g + kv_y^2} dv_y = -dt. \quad (2.68)$$

Hacemos la sustitución  $u = v_y \sqrt{k}$ , de modo que  $kv_y^2 = u^2$ . Entonces:

$$\frac{1}{g + u^2} du = -\sqrt{k} dt. \quad (2.69)$$

La integral del lado izquierdo es:

$$\int \frac{1}{g + u^2} du = \frac{1}{\sqrt{g}} \arctan \left( \frac{u}{\sqrt{g}} \right). \quad (2.70)$$

Por lo tanto:

$$\frac{1}{\sqrt{g}} \arctan \left( v_y \sqrt{\frac{k}{g}} \right) = -\sqrt{gk} t + C_2 \Rightarrow v_y \sqrt{\frac{k}{g}} = \tan \left( -\sqrt{gk} t + C_2 \right) \quad (2.71)$$

Resolviendo para  $v_y(t)$ :

$$v_y(t) = \sqrt{\frac{g}{k}} \tan \left( -\sqrt{gk} t + C_2 \right). \quad (2.72)$$

Sabemos que la condición inicial es:  $v_y(0) = v_{y0} = \sqrt{\frac{g}{k}} \tan C \Rightarrow \tan C = v_{y0} \sqrt{\frac{k}{g}}$

Por lo tanto,

$$C_2 = \arctan \left( v_{y0} \sqrt{\frac{k}{g}} \right) \quad (2.73)$$

Sustituyendo el valor de la constante de integración, obtenemos:

$$v_y(t) = \sqrt{\frac{g}{k}} \tan \left( -\sqrt{gk} t + \arctan \left( v_{y0} \sqrt{\frac{k}{g}} \right) \right). \quad (2.74)$$

Para la posición  $y(t)$ , integramos:

$$y(t) = \int v_y(t) dt = \int \sqrt{\frac{g}{k}} \tan \left( -\sqrt{gk} t + \arctan \left( v_{y0} \sqrt{\frac{k}{g}} \right) \right) dt \quad (2.75)$$

Definimos el cambio de variable:

$$\theta(t) = -\sqrt{gk} t + \arctan \left( v_{y0} \sqrt{\frac{k}{g}} \right) \quad (2.76)$$

Entonces:

$$\frac{d\theta}{dt} = -\sqrt{gk} \quad (2.77)$$

$$dt = \frac{d\theta}{-\sqrt{gk}} \quad (2.78)$$

Reescribiendo la integral:

$$y(t) = \sqrt{\frac{g}{k}} \int \tan(\theta) dt = -\frac{1}{k} \int \tan(\theta) d\theta \quad (2.79)$$

La integral de la tangente es:

$$\int \tan(\theta) d\theta = \ln |\sec(\theta)| \quad (2.80)$$

Por lo tanto:

$$y(t) = -\frac{1}{k} \ln |\sec(\theta(t))| + C \quad (2.81)$$

Aplicando la condición inicial  $y(0) = 0$ :

$$y(0) = -\frac{1}{k} \ln \left( \sqrt{1 + v_{y0}^2 \frac{k}{g}} \right) + C = 0 \quad (2.82)$$

De donde se obtiene:

$$C = \frac{1}{k} \ln \left( \sqrt{1 + v_{y0}^2 \frac{k}{g}} \right) \quad (2.83)$$

La posición queda entonces:

$$y(t) = -\frac{1}{k} \ln \left| \sec \left( -\sqrt{gk} t + \arctan \left( v_{y0} \sqrt{\frac{k}{g}} \right) \right) \right| + \frac{1}{k} \ln \left( \sqrt{1 + v_{y0}^2 \frac{k}{g}} \right) \quad (2.84)$$

Simplificando:

$$y(t) = \frac{1}{k} \ln \left( \frac{\sqrt{1 + v_{y0}^2 \frac{k}{g}}}{\sec \left( -\sqrt{gk} t + \arctan \left( v_{y0} \sqrt{\frac{k}{g}} \right) \right)} \right) \quad (2.85)$$

Utilizando que  $\sec(\theta) = \frac{1}{\cos(\theta)}$ , se tiene:

$$y(t) = \frac{1}{k} \ln \left( \sqrt{1 + v_{y0}^2 \frac{k}{g}} \cdot \cos \left( \sqrt{gk} t - \arctan \left( v_{y0} \sqrt{\frac{k}{g}} \right) \right) \right) \quad (2.86)$$

## 2.4. Formulación del modelo completo

Mientras va avanzando el modelo, se toman menos simplificaciones, lo que afecta considerablemente la complejidad de resolución del sistema de ecuaciones asociadas al sistema. Sin embargo, para llevar a cabo una simulación que se acerque lo más posible al problema físico, es vital tomar en cuenta los fenómenos más relevantes, es decir, aquellos que afecten considerablemente su trayectoria de vuelo.

Con lo anterior en mente, en esta sección se desarrolla el sistema de ecuaciones diferenciales ordinarias completo que se considera rige la dinámica lineal y rotacional de los vehículos suborbitales en este trabajo.

Para la simulación de la trayectoria del cohete se necesita, además de la descripción cinemática, una formulación matemática de su dinámica. Dichas formulaciones se complementan con los análisis presentados principalmente en los artículos de Aspire Space “*Rocket aerodynamics*” [Aspirespace, 2011b] y “*A dynamic stability analysis rocket simulator*” [Aspirespace, 2011a].

A diferencia de otros simuladores, el modelo que se presenta considera el cohete como un cuerpo rígido, y no como una partícula de masa puntual. De modo que, se tiene en cuenta la posición y orientación (inclinación del cohete sobre el plano XZ) del vehículo. Así como los momentos que se generan a partir de las fuerzas correspondientes. En trabajos posteriores, se puede considerar un cuerpo elástico para conocer la respuesta estructural ante el vuelo.

Se considera el estado completo del vehículo en términos del siguiente vector, que incluye la posición, orientación y velocidades tanto lineales como rotacionales:

$$\vec{u} = \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \\ \theta \\ \dot{\theta} = \omega \end{bmatrix} \quad (2.87)$$

De donde se sigue que su derivada es el vector que contiene las componentes de las

velocidades y aceleraciones, lineales y rotacionales.

$$\frac{\partial \vec{u}}{\partial t} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ a_x \\ a_y \\ a_z \\ \dot{\theta} = \omega \\ \ddot{\theta} = \dot{\omega} \end{bmatrix} \quad (2.88)$$

#### 2.4.1. Dinámica de vuelo

Dada las fuerzas del motor, las dimensiones y geometrías del vehículo, el objetivo es conocer la trayectoria que obtiene, es decir, conocer la posición y orientación en cada instante del vuelo, así como el lugar de posible impacto. Para lo anterior, se deducen a continuación las ecuaciones diferenciales que describen el movimiento.

Para representar el modelo dinámico del problema se considera el siguiente diagrama de cuerpo libre (Figura 2.1).

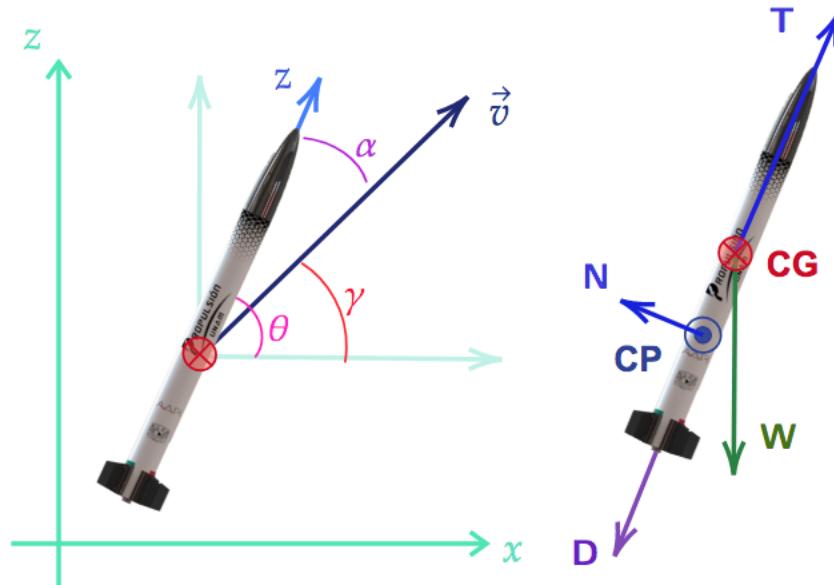


Figura 2.1: Diagrama de cuerpo libre y ángulos  
( $\alpha$ : ángulo de ataque,  $\gamma$ : ángulo de vuelo,  $\theta$ : ángulo de inclinación o “pitch”)

Para describir la trayectoria del vehículo se parte de la Tercera Ley de Newton, que nos indica que con toda acción ocurre siempre una reacción igual y contraria: o sea, las acciones mutuas de dos cuerpos siempre son iguales y dirigidas en direcciones opuestas. Lo anterior, es el principio que provoca el movimiento de un cohete. Expresado

matemáticamente como:

$$\vec{F}_{AB} = -\vec{F}_{BA},$$

donde  $\vec{F}_{AB}$  es la fuerza del motor y  $\vec{F}_{BA}$  es la fuerza que desplaza al cohete.

Recordemos ahora la Segunda Ley de Newton. Siendo  $m(t)$  la masa del vehículo que varía dependiendo del tiempo  $t$  y  $\vec{v}$  el vector de velocidad.

$$\vec{F} = m\vec{a} = m \frac{d\vec{v}}{dt}$$

El cohete se va a mover debido a las fuerzas externas que actúan sobre él

$$\vec{F} = \vec{F}_T + \vec{F}_A + \vec{F}_g,$$

donde  $\vec{F}_T$  es el vector de la fuerza de empuje (T: *Thrust*),  $\vec{F}_D$  el vector de las fuerzas aerodinámicas (A: *Aerodynamic -Drag y Normal-*) y  $\vec{F}_g$  la fuerza gravitacional debida al peso, respectivamente.

De esta forma,

$$\vec{F}_T + \vec{F}_A + \vec{F}_g = m(t) \frac{d\vec{v}}{dt}$$

Además, recordemos que la masa del sistema no es constante, pues el combustible se va consumiendo en los primeros segundos del vuelo hasta llegar a una masa constante de los componentes “secos” del vehículo, lo cual incluye todas las masas excepto la del propelente y oxidante, en caso de vehículos de combustible líquido o híbrido. Es así que,

$$m(t) = m_{seca} + m_{prop}(t), \quad (2.89)$$

donde  $m_{prop}(t) \rightarrow 0$  al consumirse el propelente.

## Ángulos utilizados

Como sabemos, el vehículo no siempre vuela en la dirección de su orientación, por lo que el análisis dinámico del vuelo del cohete requiere la definición precisa de los ángulos que describen la orientación y la trayectoria del vehículo, ya que estos determinan la dirección de aplicación de las fuerzas aerodinámicas y el comportamiento del sistema. En este trabajo se consideran tres ángulos principales, como se muestra en la Figura 2.1.

- **Ángulo de inclinación**  $\theta(t)$ : corresponde a la orientación del eje longitudinal del cohete con respecto a la horizontal del suelo. Este ángulo es función del tiempo y su valor inicial  $\theta_0$  está determinado por la inclinación de la base de lanzamiento.

$$\theta(0) = \theta_0 \quad (2.90)$$

- **Ángulo de vuelo**  $\gamma(t)$ : es el ángulo formado por el vector de velocidad relativa  $\vec{v}_{\text{rel}}$  con respecto a la horizontal. Se calcula en el plano  $XZ$  como:

$$\gamma(t) = \arctan \left( \frac{v_z}{v_x} \right) \quad (2.91)$$

- **Ángulo de ataque**  $\alpha(t)$ : representa la diferencia entre el ángulo de inclinación  $\theta(t)$  y el ángulo de vuelo  $\gamma(t)$ . Este ángulo indica qué tanto desviado está el vehículo de su dirección de vuelo y es fundamental para el cálculo de la fuerza normal aerodinámica. Se define como:

$$\alpha(t) = \theta(t) - \gamma(t) \quad (2.92)$$

### 2.4.2. Fuerzas involucradas

Como mencionamos anteriormente, durante el vuelo de un cohete, actúan diversas fuerzas que determinan su movimiento tras el lanzamiento. En esta sección se describen cada una de ellas, junto con su modelado matemático tal como fue implementado en el simulador.

#### Fuerza gravitacional

La fuerza gravitacional es uno de los componentes fundamentales en la dinámica del vuelo, ya que actúa constantemente sobre el vehículo en dirección vertical hacia la superficie terrestre. En este trabajo, se ha considerado un modelo gravitacional que varía con la altitud, pero que sigue siendo válido para trayectorias suborbitales, donde  $h \ll R_T$ , con  $R_T$  el radio de la Tierra y  $h$  la altura máxima alcanzada por el vehículo.

Se emplea una expresión basada en la ley de gravitación universal, que considera la aceleración gravitacional como función de la altitud  $h$ :

$$g(h) = \frac{G \cdot M_T}{(R_{Tierra} + h)^2}, \quad (2.93)$$

donde  $G$  es la constante de gravitación universal,  $M_T$  la masa de la Tierra y  $R_T$  el radio de la Tierra.

De la misma forma, no se considera curvatura terrestre, pues se supone una superficie plana de la Tierra apreciable a la escala del modelo. Por ende, esta fuerza se modela como un vector constante en dirección vertical descendente, cuya magnitud varía con  $h$ . Por tanto, la fuerza aplicada sobre el vehículo de masa  $m(t)$  es:

$$\vec{G}(h) = m(t) \cdot \begin{bmatrix} 0 \\ 0 \\ -g(h) \end{bmatrix} \quad (2.94)$$

Este vector se incorpora directamente en la ecuación de movimiento lineal. En el simulador,  $g(h)$  se evalúa en cada paso de integración numérica a partir de la coordenada  $z(t)$ , lo cual permite capturar el decrecimiento de la gravedad con la altitud.

### Fuerza de Empuje

La fuerza de empuje  $\vec{T}(t)$  se genera por la expulsión de gases a alta velocidad a través de la tobera del motor, de acuerdo con la tercera ley de Newton. Esta fuerza actúa sobre el centro de masa del cohete y está dirigida a lo largo de su eje longitudinal, en sentido opuesto al flujo de gases.

En este trabajo, el empuje se modela como una función del tiempo basada en datos experimentales obtenidos mediante pruebas estáticas del motor. La curva  $T_{ideal}(t)$  se obtiene a partir del registro de la fuerza neta medida sobre un banco de pruebas instrumentado. Esta curva es luego interpolada y utilizada durante la simulación.

En el simulador, la magnitud del empuje en cada instante se evalúa como:

$$T(t) = T_{ideal}(t) \cdot \epsilon_f, \quad (2.95)$$

donde  $\epsilon_f \in [0, 1]$  es un factor de eficiencia del motor, que permite ajustar el rendimiento nominal en vuelo. Este factor puede representar pérdidas por condiciones no ideales, desviaciones térmicas o rendimiento parcial de la mezcla en motores híbridos. En las simulaciones realizadas, se utiliza típicamente  $\epsilon_f = 0.95$ . Pero se permite ajustar el valor para realizar simulaciones de sensibilidad o para representar diferentes condiciones de operación del motor.

La dirección del empuje está definida por la orientación del cohete, y se proyecta en el plano de vuelo  $XZ$  de la siguiente manera:

$$\vec{T}(t) = T(t) \hat{z}_b = T(t) \cdot \begin{bmatrix} \cos \theta(t) \\ 0 \\ \sin \theta(t) \end{bmatrix} = T_{ideal}(t) \cdot \epsilon_f \cdot \begin{bmatrix} \cos \theta(t) \\ 0 \\ \sin \theta(t) \end{bmatrix}, \quad (2.96)$$

donde  $\hat{z}_b$  es la dirección del eje longitudinal del cohete y  $\theta$  es el ángulo de inclinación con respecto a la horizontal.

La curva de empuje  $T_{ideal}(t)$  se introduce a partir de un archivo de datos experimentales o de simulación que contiene valores discretos del empuje en función del tiempo. Datos que se interpolan mediante un esquema lineal, para poder evaluar el empuje en cualquier instante  $t$  durante la simulación. La función resultante es:

$$T_{ideal}(t) = \text{interp}(t, \{t_i, T_i\}) \quad (2.97)$$

, donde  $\{t_i, T_i\}$  son los puntos muestreados del empuje.

También, se define un instante de corte del motor  $t_{MECO}$  a partir del último valor no nulo del empuje. Para todos los tiempos posteriores se impone:

$$T(t > t_{MECO}) = 0, \quad (2.98)$$

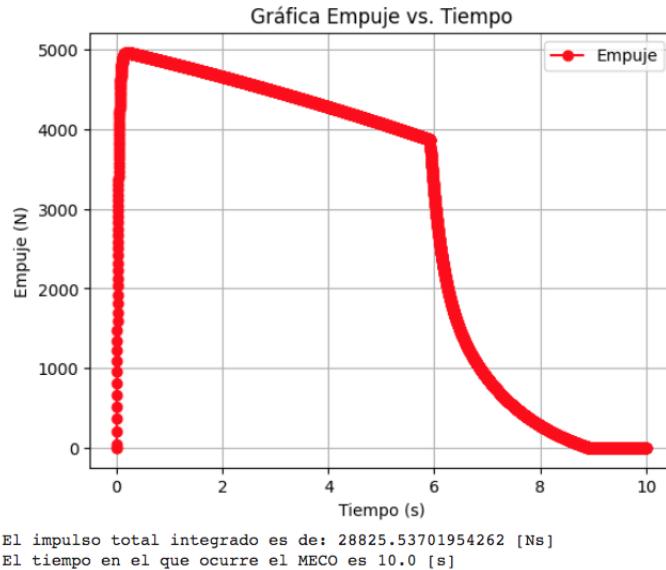


Figura 2.2: Curva de Empuje del motor "Mega Punisher"

donde *MECO* hace referencia al acrónimo en inglés "*Main Engine Cut Off*". Esto garantiza que el cohete entra en fase balística una vez agotado el combustible, respetando la duración real del funcionamiento del motor.

En esta tesis no se consideran desviaciones del eje de empuje respecto al eje del cohete (i.e., desalineación angular). Por lo tanto, no se genera ningún momento inducido por el empuje, resultando en un vector de empuje perfectamente alineado con el eje longitudinal.

En trabajos futuros, puede extenderse el simulador para incorporar control vectorial de empuje (*Thrust Vectoring Control*), lo cual permitiría modelar momentos generados por pequeñas desviaciones angulares del flujo. Así mismo, en vehículos de motores híbridos o líquidos la modelación del empuje involucra otras complejidades que salen del alcance de este trabajo.

### Efecto del viento

El viento no se modela como una fuerza directa, sino que altera la velocidad relativa del cohete respecto al aire. En la simulación se utiliza un modelo 3D de viento con magnitud y dirección variables.

$$\vec{v}_{rel} = \vec{v}_{cohete} - \vec{v}_{viento} \quad (2.99)$$

Esto modifica tanto el cálculo del arrastre como el de la fuerza normal y está representado en el siguiente diagrama:

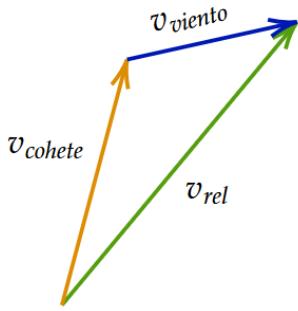


Figura 2.3: Vectores de velocidad del cohete, velocidad del viento y velocidad relativa

### Arrastre

La fuerza de arrastre es una de las principales fuerzas aerodinámicas que actúan sobre el cohete durante el vuelo atmosférico. Esta fuerza representa la resistencia que ofrece el aire al movimiento del vehículo, actuando siempre sobre el centro de presión en dirección opuesta a la velocidad relativa del cohete con respecto al viento.

$$\vec{D} = -\frac{1}{2}\rho(z)|\vec{v}_{rel}|^2 Cd(M)A \cdot \hat{v}_{rel}, \quad (2.100)$$

donde:

- $\rho(z)$  es la densidad del aire a la altitud  $z$  calculada usando un modelo atmosférico estratificado,
- $Cd(M)$  es el coeficiente de arrastre como función del número de Mach
- $A$  es el área de sección transversal del cohete.
- $\vec{v}_{rel}$  es el vector unitario en la dirección del flujo relativo, definido anteriormente en (Ec. 2.99)

En coordenadas del simulador, se proyecta sobre el plano  $XZ$ , y se asume  $D_y = 0$ .

Para cohetes de gran escala, la fuerza de arrastre es de orden mucho menor comparada con las fuerzas de empuje y peso, por lo cual se ignora para simplificar los cálculos. Sin embargo, para cohetes suborbitales el vuelo se realiza dentro de la atmósfera, donde es importante considerar dicha fuerza.

Factores como la rugosidad de la superficie, variaciones de manufactura, entre otros afectan directamente su magnitud, por lo que es difícil conocer con exactitud su valor. Sin embargo, modelar esta fuerza no es el propósito de este trabajo.

### Variación del coeficiente de arrastre

Durante el vuelo, el cohete experimenta una aceleración significativa que puede llevárselo a atravesar distintos regímenes de velocidad, clasificados según el número de

Mach:

$$M = \frac{|\vec{v}_{\text{rel}}|}{c_s(z)}, \quad (2.101)$$

donde  $c_s(z)$  es la velocidad local del sonido, dependiente de la altitud y calculada mediante el modelo atmosférico ISA (Revisar el apéndice para más detalles).

En el simulador, el coeficiente se obtiene mediante interpolación de una tabla generada con el software OpenRocket, que relaciona valores de  $C_d$  con el número de Mach  $M$ . Esta curva incluye el aumento característico de arrastre en el régimen transónico ( $M \approx 1$ ), debido a la formación de ondas de choque que provocan una compresión súbita del aire, aumentando la presión y temperatura local, lo que se traduce en una mayor resistencia al avance.

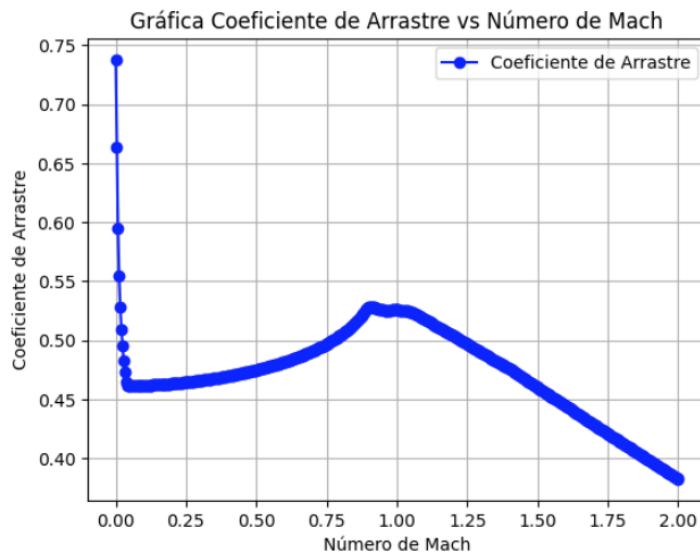


Figura 2.4: Variación del coeficiente de arrastre de acuerdo a la velocidad

Así, para calcular el arrastre correspondiente a cada instante se realiza el siguiente cálculo:

$$C_d(M) = \text{interp}(M, \text{tabla}_{C_d}) \quad (2.102)$$

Este enfoque permite capturar los efectos del aumento de arrastre en el régimen transónico y su posterior comportamiento en el flujo supersónico, sin necesidad de modelar directamente las ondas de choque.

Como trabajo futuro, se prevé el desarrollo de un módulo aerodinámico interno al simulador, que integre modelos analíticos y semiempíricos para el cálculo de los coeficientes aerodinámicos en función de la geometría del vehículo. Esto permitirá prescindir de software externo y ampliar la aplicabilidad del simulador a diversas configuraciones de diseño.

### Corrección con paracaídas activado

Si se activa el paracaídas, se añade una fuerza de arrastre adicional.

$$\vec{D}_{paracaidas} = -\frac{1}{2}\rho(z)|\vec{v}_{rel}|^2Cd_p(M)A_p \cdot \hat{v}_{rel}, \quad (2.103)$$

donde

- $Cd_p$  es el coeficiente de arrastre del paracaídas,
- $A_p$  es el área efectiva del paracaídas.

De donde,

$$\vec{D} = \vec{D}_{cohete} + \vec{D}_{paracaidas} \quad (2.104)$$

### Fuerza Normal

La fuerza normal  $\vec{N}$  es una fuerza perpendicular al eje del cohete que aparece cuando el cohete no está perfectamente alineado, es decir, tiene un ángulo de ataque  $\alpha > 0$  respecto al flujo de aire.

Esta desalineación se mide mediante el ángulo de ataque  $\alpha$  y provoca una fuerza que tiene un efecto estabilizador (o desestabilizador) dependiendo de la ubicación relativa del centro de presión y el centro de gravedad. Así, el punto de aplicación de la fuerza normal es el centro de presión y su dirección varía de acuerdo al ángulo de ataque  $\alpha$ .

Esta fuerza actúa sobre el plano  $XZ$ , perpendicular al arrastre, y se calcula como:

$$\vec{N} = \frac{1}{2}\rho(z)|\vec{v}_{rel}|^2C_N(\alpha)A \cdot \hat{n}, \quad (2.105)$$

donde

- $\rho(z)$  es la densidad del aire a la altitud  $z$ , calculada de acuerdo a la ISA (Revisar el apéndice para más detalles)
- $A$  es el área de referencia del cuerpo,
- $C_N(\alpha)$  es el coeficiente de fuerza normal,
- $\hat{n}$  es el vector unitario perpendicular al flujo, orientado por el signo de  $\alpha$

En este trabajo, se adopta una formulación simple del coeficiente como función del ángulo de ataque:

$$C_N(\alpha) = C_{N,max} \cdot |\sin(\alpha)| \quad (2.106)$$

Esta expresión capta adecuadamente el comportamiento de cuerpos esbeltos a ángulos de ataque pequeños y moderados. Entonces, el valor de  $C_{N,max}$  se fija como parámetro del vehículo, calibrado para representar el efecto de las aletas.

En el simulador, la fuerza normal se descompone en coordenadas cartesianas y se calcula en cada paso de tiempo:

$$N_x = N \cdot (-\sin \alpha) \quad (2.107)$$

$$N_y = 0 \quad (2.108)$$

$$N_z = N \cdot \cos \alpha \quad (2.109)$$

Esta fuerza no solo produce aceleración transversal sino que genera un momento aerodinámico, del que se hablará más adelante.

### 2.4.3. Consideraciones angulares

#### Momentos aerodinámicos

Además de las fuerzas que actúan sobre el cohete, es fundamental considerar los momentos que estas fuerzas inducen respecto al centro de gravedad, ya que determinan el comportamiento angular del vehículo. En particular, las fuerzas aerodinámicas de arrastre y normal pueden generar momentos que modifican el ángulo de orientación del cohete, influenciando directamente su estabilidad.

#### Cálculo del momento neto

El momento total  $\vec{\tau}$  que actúa sobre el cohete se calcula a partir de los vectores de palanca y las fuerzas aerodinámicas aplicadas. En el simulador, se considera que estas fuerzas actúan en el centro de presión del vehículo, mientras que la rotación ocurre alrededor del centro de gravedad. De esta forma, el vector de palanca en coordenadas del cohete está dado por:

$$\vec{r}_b = \vec{CP} - \vec{CG}, \quad (2.110)$$

donde  $CP$  y  $CG$  son las posiciones del centro de presión y del centro de gravedad, respectivamente.

Este vector se transforma al sistema de coordenadas del suelo mediante una matriz de rotación  $M(\theta)$ , en función del ángulo de inclinación  $\theta$ :

$$\vec{r} = M(\theta) \cdot \vec{r}_b, \quad \text{con} \quad M(\theta) = \begin{bmatrix} \sin \theta & 0 & -\cos \theta \\ 0 & 1 & 0 \\ \cos \theta & 0 & \sin \theta \end{bmatrix} \quad (2.111)$$

El momento aerodinámico total es entonces la suma de los momentos generados por el arrastre  $\vec{D}$  y la fuerza normal  $\vec{N}$ :

$$\vec{\tau} = \vec{r} \times \vec{D} + \vec{r} \times \vec{N} \quad (2.112)$$

### Dinámica angular del cohete

La componente que interesa del momento total es la que actúa en el eje transversal  $y$ , perpendicular al plano de vuelo  $XZ$ . En el simulador, esta componente se invierte de signo debido a la convención adoptada en el sistema de coordenadas:

$$\tau_y = -(\vec{\tau})_y \quad (2.113)$$

Así, la aceleración angular del cohete se calcula mediante la segunda ley de Newton rotacional:

$$\alpha = \frac{\tau_y}{I_x}, \quad (2.114)$$

donde  $I_x$  es el momento de inercia total del cohete respecto al eje transversal, calculado como suma de los momentos de inercia de cada componente y el efecto del teorema de ejes paralelos (Revisar el apéndice para más detalles).

De esta forma, se considera el estado angular  $\theta(t)$  y la velocidad angular  $\omega(t)$  para deducir las siguientes ecuaciones rotacionales:

$$\frac{d\theta}{dt} = \omega, \quad \frac{d\omega}{dt} = \alpha \quad (2.115)$$

### Estabilidad

Cualquier simulador de vehículos requiere la determinación adecuada de su comportamiento dinámico y la comprobación de que se cumplen cualidades de vuelo seguras. Lo anterior garantiza que el vehículo no volará descontroladamente por el cielo, corriendo la posibilidad de dañar a alguien o aterrizar en algún bien material o construcción.

Como se menciona en [Tierno et al., 2012], el tema de la estabilidad y control se han tratado mayoritariamente de forma cualitativa basándose en parámetros de vehículos semejantes. Sin embargo, es de vital importancia asegurarse de que los cohetes suborbitales cumplen ciertas recomendaciones de estabilidad y controlabilidad, especialmente en la fase propulsiva y durante la transición hacia el apogeo.

Para esto, tomemos en cuenta que un cohete se considera **estáticamente estable** cuando, tras una perturbación angular, el vehículo genera un momento aerodinámico que tiende a restaurar su orientación inicial. Este tipo de estabilidad implica que existe una tendencia inmediata a regresar al equilibrio angular. Por otro lado, se habla de **estabilidad dinámica** cuando dicho retorno no ocurre instantáneamente, pero el movimiento oscilatorio inducido por la perturbación se atenúa con el tiempo, llevando eventualmente al restablecimiento de la orientación inicial.

La condición más relevante en este contexto es la relación geométrica entre el centro de presión ( $CP$ ) y el centro de gravedad ( $CG$ ). En particular:

- Si el centro de presión se ubica por delante del centro de gravedad, cualquier perturbación angular genera un momento desestabilizador, incrementando la desviación del eje de vuelo.
- Si el centro de presión está por detrás del centro de gravedad, se genera un momento restaurador que favorece el retorno al eje de vuelo original.

Esto se ilustra en la Figura 2.5, donde se representa gráficamente el sentido del momento generado por una perturbación angular, en configuraciones estables e inestables.



Figura 2.5: Momento estabilizador en configuración no estable y estable

Para cuantificar la estabilidad estática, se utiliza el concepto de margen de estabilidad o calibre, definido como:

$$\text{calibre}(t) = \frac{z_{CG} - z_{CP}}{D}, \quad (2.116)$$

donde  $z_{CP}$  y  $z_{CG}$  son las coordenadas longitudinales del centro de presión y de gravedad, respectivamente, y  $D$  es el diámetro máximo del fuselaje del cohete.

Un calibre positivo indica que el  $CP$  está por detrás del  $CG$ , condición necesaria para la estabilidad estática, mientras que un calibre negativo indica inestabilidad. Según la literatura especializada (ver por ejemplo [Tim, 1997] o [Niskanen, 2013]), se considera que un calibre adecuado debe estar en el rango de 1 a 2 calibres, es decir, el centro de presión debe ubicarse entre una y dos veces el diámetro del cuerpo por detrás del centro de gravedad. Este criterio proporciona un buen compromiso entre estabilidad y maniobrabilidad.

Cabe mencionar que el centro de presión es una función del ángulo de ataque y puede desplazarse durante el vuelo. En particular, se ha observado que el  $CP$  tiende a moverse hacia adelante cuando el ángulo de ataque aumenta, lo que puede comprometer temporalmente la estabilidad si no se considera adecuadamente en el diseño.

Bajo condiciones de vuelo estables, el comportamiento del cohete frente a una perturbación angular se caracteriza como un movimiento oscilatorio amortiguado (*damped oscillatory motion*), compuesto por dos momentos principales:

1. Un momento restaurador  $M_s$ , asociado al desequilibrio entre  $CG$  y  $CP$ .
2. Un momento de amortiguamiento  $Ma$ , generado por la interacción aerodinámica, que reduce progresivamente la amplitud de la oscilación.

Ambos efectos contribuyen a que el vehículo recupere su trayectoria nominal incluso en presencia de perturbaciones externas, como ráfagas de viento o irregularidades en la propulsión.

#### 2.4.4. Ecuaciones

Así, de las secciones anteriores se obtiene que la dinámica general del cohete está dada por las siguientes ecuaciones diferenciales ordinarias temporales referidas a la aceleración y velocidad de la parte lineal y angular del vehículo. Posteriormente, haciendo uso de un cambio de variable, se transforma en el siguiente sistema de ecuaciones diferenciales ordinarias de primer orden.

##### PARTE LINEAL (vectorial): ECUACIONES DE NEWTON

$$\begin{aligned} \vec{a}(t) &= \frac{d\vec{v}}{dt} = \frac{1}{m(t)} \left[ m(t) \begin{bmatrix} 0 \\ 0 \\ -g(h) \end{bmatrix} + T_{\text{ideal}}(t) \cdot \epsilon_f \begin{bmatrix} \cos \theta(t) \\ 0 \\ \sin \theta(t) \end{bmatrix} \right. \\ &\quad \left. - \frac{1}{2} \rho(z) |\vec{v}_{\text{rel}}|^2 C_d(M) A \cdot \hat{v}_{\text{rel}} + \frac{1}{2} \rho(z) |\vec{v}_{\text{rel}}|^2 C_N(\alpha) A \cdot \hat{n} \right], \end{aligned} \quad (2.117)$$

$$\vec{v}(t) = \int \vec{a}(t) dt + \vec{v}_0,$$

$$\vec{x}(t) = \int \vec{v}(t) dt + \vec{x}_0.$$

##### PARTE ANGULAR (escalar): ECUACIÓN DE EULER

$$\begin{aligned} \alpha(t) &= \frac{d\omega}{dt} = \frac{-1}{I_x} \left[ \left( \mathbf{M}(\theta) \cdot (\vec{C}P - \vec{C}G) \right) \times \left( -\frac{1}{2} \rho(z) |\vec{v}_{\text{rel}}|^2 C_d(M) A \cdot \hat{v}_{\text{rel}} \right) \right. \\ &\quad \left. + \left( \mathbf{M}(\theta) \cdot (\vec{C}P - \vec{C}G) \right) \times \left( \frac{1}{2} \rho(z) |\vec{v}_{\text{rel}}|^2 C_N(\alpha) A \cdot \hat{n} \right) \right]_y, \end{aligned} \quad (2.118)$$

$$\omega(t) = \int \alpha(t) dt + \omega_0,$$

$$\theta(t) = \int \omega(t) dt + \theta_0,$$

Cabe destacar que las ecuaciones translacionales están desacopladas de las ecuaciones rotacionales. De modo que podemos resolver un sistema (la posición del centro de gravedad) sin resolver el otro (la variación en la orientación del vehículo).

Como resultado de la simplificación anterior, se requieren menos condiciones iniciales para resolver el sistema.

## Características del sistema

El sistema de ecuaciones diferenciales ordinarias anterior conformado por las ecuaciones en (2.117) y (2.118) presenta una serie de características que dificultan la obtención de soluciones analíticas y condicionan la elección y desempeño de los métodos numéricos usados para resolverlas. A continuación, se detallan las propiedades más relevantes del sistema:

1. **No linealidad** El sistema dinámico incluye múltiples términos no lineales, cuya fuente principal proviene de:

- La fuerza aerodinámica: La resistencia del aire depende cuadráticamente de la velocidad  $\vec{D} \propto \vec{v}^2$ , y los coeficientes aerodinámicos pueden ser función del ángulo de ataque, que a su vez depende de la orientación y la velocidad del cohete.
- La variación de masa: Debido al consumo de propelente, la masa del cohete disminuye a lo largo del tiempo, afectando la aceleración según la Segunda Ley de Newton.

Esta no linealidad implica que pequeñas variaciones en las condiciones iniciales pueden generar trayectorias notablemente distintas, por lo que los métodos numéricos deben ser robustos y estables ante tales variaciones.

2. **Acoplamiento** Las variables que conforman el vector de estado  $u(t)$  están acopladas, es decir, la evolución de cada componente en las ecuaciones rotacionales y translacionales están conectadas a través de las fuerzas y momentos aerodinámicos. Por ejemplo:

- La posición depende de la velocidad.
- La velocidad está determinada por las fuerzas que dependen de la masa, orientación y posición.
- El ángulo de inclinación afecta la dirección del empuje y, por tanto, la aceleración.

Este acoplamiento impide resolver cada ecuación de manera independiente, lo que obliga a considerar el sistema completo en cada paso de integración.

3. **Possible rigidez** El sistema puede presentar comportamiento rígido en ciertas fases de la trayectoria. Un sistema rígido se caracteriza por contener compo-

nentes con escalas de tiempo muy diferentes, lo que obliga a utilizar pasos de integración muy pequeños para garantizar la estabilidad numérica, incluso si las soluciones varían lentamente. En el caso particular de la dinámica de vuelo de un cohete, la rigidez puede manifestarse durante:

- La fase de ignición: Cuando el motor se enciende, se produce un cambio abrupto en la aceleración.
- Eventos discontinuos: Como el apagado del motor o la separación de etapas (si existiera).
- Pequeñas perturbaciones: En vuelos de alta precisión, pequeñas variaciones pueden inducir oscilaciones rápidas en la inclinación.

Aunque el sistema completo no es extremadamente rígido, estas situaciones locales deben ser tratadas con métodos que permitan adaptarse dinámicamente para mantener la precisión sin comprometer la estabilidad.

4. **Sensibilidad a condiciones iniciales** La dinámica del cohete también presenta sensibilidad a las condiciones iniciales, fenómeno característico de sistemas no lineales. Esto significa que errores pequeños en las condiciones de partida pueden amplificarse a lo largo del tiempo, afectando significativamente la trayectoria calculada.

Este aspecto exige la utilización de métodos numéricos con control de error y, en muchos casos, la realización de análisis de sensibilidad para garantizar la confiabilidad de la simulación.

### Condiciones iniciales y estado del sistema

El sistema formado por (Ec. 2.117) y (Ec. 2.118) de ecuaciones diferenciales acopladas debe resolverse partiendo de un estado inicial  $u_0 = (x_0, y_0, z_0, v_{x0}, v_{y0}, v_{z0}, \theta_0, \dot{\theta}_0)$ , que en nuestro caso representa la posición, velocidad, ángulo y velocidad angular inicial del lanzamiento. En general, dado que el vehículo parte del reposo tendremos al estado inicial como  $u_0 = (0, 0, 0, 0, 0, 0, \theta_0, 0)$ , donde el lugar de lanzamiento se considera el origen del sistema coordenado.

### Discretización temporal e integración

Dado que no es posible obtener una solución analítica cerrada para este sistema de ecuaciones no lineales, se recurre a métodos numéricos de integración. En el simulador, se ofrece una variedad de esquemas, de los cuales se hablará a detalle en el siguiente capítulo.

En cada paso temporal  $dt$ , se calcula la aceleración lineal y angular, se actualizan las derivadas de primer orden y se construye el nuevo estado del sistema.

Este procedimiento se repite de forma iterativa hasta alcanzar el tiempo máximo de simulación o hasta que el cohete impacte el suelo (es decir, cuando  $z(t) < 0$ ).

## 3 Métodos numéricos

El análisis y simulación de la dinámica de vuelo de un cohete suborbital requiere la resolución de sistemas de ecuaciones diferenciales ordinarias no lineales, así como la adecuada inclusión de información proveniente de tablas de datos experimentales o simuladas. Estas tareas no pueden resolverse de forma exacta mediante métodos analíticos, debido a la complejidad y naturaleza del sistema dinámico. Por esta razón, es necesario recurrir a técnicas numéricas que permitan obtener soluciones aproximadas dentro de un margen de error controlado y con un costo computacional razonable.

Los métodos numéricos son herramientas matemáticas diseñadas para transformar operaciones complejas en un conjunto de cálculos aritméticos simples, que pueden ser implementados eficientemente en computadoras. Su principal ventaja radica en la capacidad de resolver problemas que no admiten solución analítica o cuya resolución exacta sería inviable por su alto costo computacional, permitiendo así abordar problemas de dinámica, fluidos, estructuras y sistemas complejos en general.

En este capítulo se presentan los fundamentos teóricos y la implementación práctica de los métodos numéricos utilizados en este trabajo, abarcando dos elementos clave:

- La interpolación lineal de datos tabulados, utilizada para la obtención de parámetros relevantes del sistema, como la fuerza de empuje y la variación de masa o de coeficiente de arrastre, a partir de información discreta.
- Los métodos numéricos de integración empleados para la resolución de sistemas de ecuaciones diferenciales ordinarias, incluyendo un análisis de su precisión, estabilidad, eficiencia y aplicabilidad.

En la primera parte del capítulo, se aborda el procedimiento de interpolación lineal aplicado a tablas de datos, pues es necesario disponer de dichos datos a distintos instantes de tiempo, determinados por el esquema de integración utilizado. Para este propósito, se emplea un método de interpolación lineal que garantiza una aproximación adecuada y eficiente de los datos, evitando fluctuaciones no deseadas y asegurando la coherencia con la dinámica simulada.

Posteriormente, se presentan los métodos numéricos de integración utilizados para la solución de las ecuaciones diferenciales ordinarias. Se incluyen métodos explícitos, adaptativos e implícitos y su respectivas características principales.

El capítulo culmina con un análisis comparativo y una validación rigurosa de los métodos seleccionados, utilizando casos dinámicos simples con solución analítica desarrollada en el capítulo anterior. Este análisis permitió identificar las ventajas y limitaciones de cada método, lo que facilitó la elección del integrador más adecuado para las simulaciones posteriores.

La correcta aplicación de las técnicas numéricas descritas en este capítulo constituye un elemento esencial para garantizar la precisión, estabilidad y eficiencia de la simulación de la dinámica de vuelo desarrollada en este trabajo.

### 3.1. Interpolación lineal

La interpolación lineal es una técnica numérica utilizada para aproximar el valor de una función en un intervalo determinado, a partir de un conjunto discreto de datos conocidos. Dado un conjunto de  $n$  datos discretos de la forma  $(x_i, f(x_i))$ , la interpolación lineal consiste en estimar el valor de la función  $f(x)$  para un valor de  $x$  perteneciente al intervalo  $[x_i, x_{i+1}]$ , mediante la ecuación de la recta que une los puntos  $(x_i, f(x_i))$  y  $(x_{i+1}, f(x_{i+1}))$ .

La expresión matemática de la interpolación lineal es:

$$f(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i), \quad (3.1)$$

donde  $x_i \leq x \leq x_{i+1}$ .

Este método presenta un error de interpolación proporcional al cuadrado de la distancia entre los puntos de la tabla, es decir, el error es de orden  $\mathcal{O}(h^2)$ , donde  $h = x_{i+1} - x_i$ . [Burden and Faires, 2021]

#### 3.1.1. Aplicación a tablas de datos

En el contexto de este trabajo, la interpolación lineal se emplea para aproximar los valores de variables clave del simulador, tales como la fuerza de empuje, la variación de masa y de coeficiente de arrastre, a partir de tablas de datos discretos. Dichas tablas, usualmente en formato .csv, pueden obtenerse de forma experimental o mediante simulaciones específicas del motor, y se consideran como información de entrada al simulador de vuelo.

Debido a que el esquema de integración numérica utilizado determina instantes de tiempo variables y, en el caso de métodos adaptativos, un tamaño de paso dinámico, es necesario conocer el valor de las variables mencionadas en tiempos intermedios que no necesariamente coinciden con los puntos discretos de las tablas. Para ello, se recurre a la interpolación lineal, permitiendo calcular los valores requeridos de manera eficiente y con un error controlado.

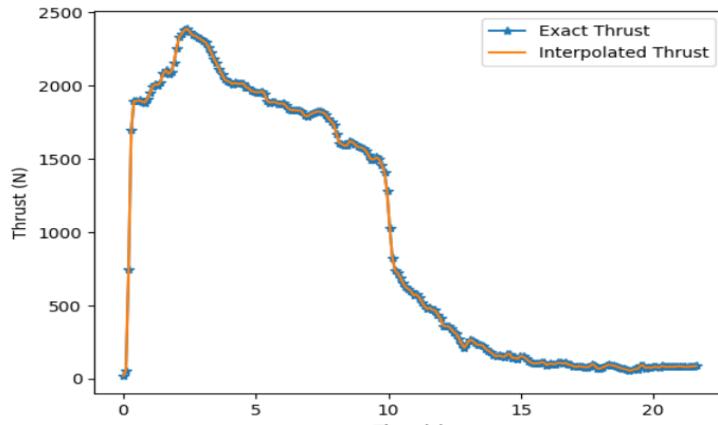
La utilización de este método es especialmente adecuada considerando las siguientes razones:

- La interpolación lineal es computacionalmente eficiente y sencilla de implementar.
- Los datos tabulados presentan suficiente resolución temporal, por lo que la variación de las variables entre dos puntos consecutivos puede asumirse lineal sin introducir un error significativo.
- El objetivo de la interpolación es proporcionar un valor razonablemente preciso en cada paso de integración, evitando la necesidad de recurrir a esquemas de interpolación más complejos que podrían incrementar el costo computacional sin un beneficio sustancial.

Un ejemplo de la aplicación de la interpolación lineal se presenta en la Figura 3.1, donde se muestra la aproximación de la fuerza de empuje a partir de los datos discretos de la tabla .

time	thrust
0	0
0.001	0
0.002	47.867317
0.003	208.91473
0.004	365.87758
0.005	518.60142
0.006	666.9864
0.007	811.00601
0.008	950.73973
0.009	1086.2308

(a) Captura parcial del archivo csv



(b) Datos reales y la interpolación realizada

Figura 3.1: Interpolación de curva de empuje

Como se observa, la interpolación lineal permite obtener un perfil continuo y coherente de la variable interpolada, preservando la tendencia de los datos originales y garantizando su disponibilidad en cualquier instante requerido por el integrador numérico.

### 3.1.2. Implementación computacional

La interpolación lineal fue implementada en Python utilizando la función `interp1d` de la librería SciPy. Este procedimiento se integró al simulador de vuelo para proporcionar, en tiempo de ejecución, los valores requeridos por el integrador, asegurando la coherencia de los datos y evitando la necesidad de discretizar a priori el intervalo temporal de la simulación.

## 3.2. Solución numérica de ecuaciones diferenciales ordinarias

Existen múltiples técnicas para la integración de EDO, entre las que se encuentran los métodos de un solo paso y los de múltiples pasos. Si bien los métodos de múltiples pasos, como los de Adams-Bashforth y Adams-Moulton, son ampliamente utilizados en ciertos contextos debido a su bajo costo computacional por paso, no fueron considerados en este trabajo. Estos métodos presentan importantes limitaciones frente a las características del sistema dinámico modelado.

En particular, la dinámica de vuelo de un cohete presenta no linealidad, rigidez localizada y sensibilidad a las condiciones iniciales, condiciones que afectan negativamente la estabilidad y precisión de los métodos multistep explícitos. Además, estos métodos presentan dificultades para implementar control adaptativo del error, lo cual resulta fundamental para el objetivo de esta simulación. Por estas razones, su inclusión no aporta ventajas relevantes en el contexto del simulador desarrollado.

En cambio, la selección de los métodos numéricos considerados para la simulación responde a un análisis cuidadoso de los requisitos y desafíos que presenta el problema. Se han incorporado métodos explícitos de paso fijo, como el método de Euler, Runge-Kutta de cuarto orden (RK4), con el objetivo de establecer una base de comparación que evidencie las limitaciones de los esquemas simples frente a la dinámica compleja del sistema.

Asimismo, se han seleccionado métodos explícitos adaptativos, como Dormand-Prince 8(53) (DOPRI853) y Runge-Kutta-Fehlberg 4(5) (RKF45), que permiten un control automático del error local mediante la variación dinámica del paso temporal.

Los métodos explícitos calculan el estado futuro del sistema directamente a partir de la información actual, mientras que los métodos implícitos requieren resolver una ecuación que involucra el valor futuro. Por lo que, para abordar situaciones donde la rigidez del sistema compromete la estabilidad de los métodos explícitos, se han incluido métodos implícitos y automáticos como la fórmula de diferenciación hacia atrás (BDF) y el algoritmo LSODA.

Esta sección tiene por objetivo exponer los fundamentos teóricos y las características principales de los 6 métodos numéricos seleccionados [Butcher, 2016], con el propósito de establecer criterios objetivos para su comparación y eventual implementación en el simulador desarrollado.

### 3.2.1. Métodos no adaptativos

Los métodos no adaptativos emplean un paso de integración constante a lo largo de toda la simulación. Aunque son más simples de implementar y permiten un control

más directo sobre la discretización temporal, su eficiencia y precisión pueden verse comprometidas en sistemas con dinámicas variables o transiciones rápidas.

### Método de Euler

El método de Euler es el esquema numérico más sencillo para la integración de EDO. Utiliza una aproximación lineal de la solución, basada en la pendiente evaluada en el instante actual.

A partir del sistema:

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{f}(t, \mathbf{y}(t)), \quad \mathbf{y}(t_0) = \mathbf{y}_0 \quad (3.2)$$

el método de Euler actualiza la solución como:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \mathbf{f}(t_n, \mathbf{y}_n), \quad (3.3)$$

donde  $h$  es el tamaño del paso temporal y  $\mathbf{y}_n$  la aproximación en  $t_n$ .

### Características principales

- Orden de precisión: 1.
- Método explícito.

### Método de Runge-Kutta de cuarto orden (RK4)

El método RK4 mejora la precisión al considerar cuatro evaluaciones de la función dinámica por paso, calculando una media ponderada de las pendientes.

El esquema de integración es:

$$\mathbf{k}_1 = \mathbf{f}(t_n, \mathbf{y}_n) \quad (3.4)$$

$$\mathbf{k}_2 = \mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_1\right) \quad (3.5)$$

$$\mathbf{k}_3 = \mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_2\right) \quad (3.6)$$

$$\mathbf{k}_4 = \mathbf{f}(t_n + h, \mathbf{y}_n + h\mathbf{k}_3) \quad (3.7)$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{6} (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \quad (3.8)$$

### Características principales

- Orden de precisión: 4.
- Método explícito.

### 3.2.2. Métodos adaptativos explícitos

Los métodos adaptativos ajustan dinámicamente el tamaño del paso temporal a lo largo de la integración, con el objetivo de mantener el error numérico dentro de un umbral de tolerancia predefinido. [Press et al., 2007]

**Métodos embebidos:** Utilizan un par de esquemas numéricos de diferente orden dentro de la misma integración, lo que permite estimar el error local sin necesidad de cálculos adicionales significativos.

Esta estrategia es fundamental en los métodos adaptativos, ya que permite ajustar dinámicamente el paso temporal para mantener la precisión deseada con eficiencia computacional. Permite decidir de forma automática si se acepta el resultado o se debe repetir el paso con un tamaño ajustado.

#### Método de Runge-Kutta-Fehlberg 4(5) (RKF45)

El método RKF45 es un esquema de Runge-Kutta adaptativo que calcula simultáneamente dos soluciones de orden 4 y 5. La diferencia entre ambas se utiliza para estimar el error local y ajustar dinámicamente el paso de integración.

A partir de:

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{f}(t, \mathbf{y}(t)) \quad (3.9)$$

el método calcula:

$$\mathbf{k}_1 = h\mathbf{f}(t_n, \mathbf{y}_n) \quad (3.10)$$

$$\mathbf{k}_2 = h\mathbf{f}\left(t_n + \frac{1}{4}h, \mathbf{y}_n + \frac{1}{4}\mathbf{k}_1\right) \quad (3.11)$$

$$\mathbf{k}_3 = h\mathbf{f}\left(t_n + \frac{3}{8}h, \mathbf{y}_n + \frac{3}{32}\mathbf{k}_1 + \frac{9}{32}\mathbf{k}_2\right) \quad (3.12)$$

$$\mathbf{k}_4 = h\mathbf{f}\left(t_n + \frac{12}{13}h, \mathbf{y}_n + \frac{1932}{2197}\mathbf{k}_1 - \frac{7200}{2197}\mathbf{k}_2 + \frac{7296}{2197}\mathbf{k}_3\right) \quad (3.13)$$

$$\mathbf{k}_5 = h\mathbf{f}\left(t_n + h, \mathbf{y}_n + \frac{439}{216}\mathbf{k}_1 - 8\mathbf{k}_2 + \frac{3680}{513}\mathbf{k}_3 - \frac{845}{4104}\mathbf{k}_4\right) \quad (3.14)$$

$$\mathbf{k}_6 = h\mathbf{f}\left(t_n + \frac{1}{2}h, \mathbf{y}_n - \frac{8}{27}\mathbf{k}_1 + 2\mathbf{k}_2 - \frac{3544}{2565}\mathbf{k}_3 + \frac{1859}{4104}\mathbf{k}_4 - \frac{11}{40}\mathbf{k}_5\right) \quad (3.15)$$

La solución de orden 4 es:

$$\mathbf{y}_{n+1}^{(4)} = \mathbf{y}_n + \frac{25}{216}\mathbf{k}_1 + \frac{1408}{2565}\mathbf{k}_3 + \frac{2197}{4104}\mathbf{k}_4 - \frac{1}{5}\mathbf{k}_5 \quad (3.16)$$

La solución de orden 5 es:

$$\mathbf{y}_{n+1}^{(5)} = \mathbf{y}_n + \frac{16}{135}\mathbf{k}_1 + \frac{6656}{12825}\mathbf{k}_3 + \frac{28561}{56430}\mathbf{k}_4 - \frac{9}{50}\mathbf{k}_5 + \frac{2}{55}\mathbf{k}_6 \quad (3.17)$$

El error local estimado es:

$$\mathbf{e}_{n+1} = \mathbf{y}_{n+1}^{(5)} - \mathbf{y}_{n+1}^{(4)} \quad (3.18)$$

### Características principales

- Orden de precisión: 4(5).
- Método explícito y adaptativo.
- Control automático del error local.
- No adecuado para sistemas rígidos.

### Método de Dormand-Prince 8(5,3) (DOPRI853)

El método Dormand-Prince 8(5,3) [Dormand and Prince, 1980] es un esquema adaptativo explícito de alto orden basado en la familia de métodos de Runge-Kutta embebidos. Utiliza una solución principal de orden 8 y dos soluciones embebidas de orden 5 y 3 para estimar el error local y controlar el paso de integración. Su diseño busca maximizar la precisión de la solución de orden 8, haciendo especial énfasis en su eficiencia en problemas no rígidos.

DOPRI853 calcula las siguientes etapas intermedias:

$$k_1 = h\mathbf{f}(t_n, \mathbf{y}_n) \quad (3.19)$$

$$k_2 = h\mathbf{f}\left(t_n + \frac{1}{18}h, \mathbf{y}_n + \frac{1}{18}k_1\right) \quad (3.20)$$

$$k_3 = h\mathbf{f}\left(t_n + \frac{1}{12}h, \mathbf{y}_n + \frac{1}{48}k_1 + \frac{1}{16}k_2\right) \quad (3.21)$$

$$k_4 = h\mathbf{f}\left(t_n + \frac{1}{8}h, \mathbf{y}_n + \frac{1}{32}k_1 + \frac{0}{0}k_2 + \frac{3}{32}k_3\right) \quad (3.22)$$

$$k_5 = h\mathbf{f}\left(t_n + \frac{5}{16}h, \mathbf{y}_n + \frac{5}{1152}k_1 + \frac{0}{0}k_2 + \frac{27}{640}k_3 + \frac{125}{384}k_4\right) \quad (3.23)$$

⋮

Debido a la extensión del método (12 etapas con coeficientes complejos), el resto de las constantes se presenta en el Anexo. La solución de orden 8 se calcula mediante una combinación lineal de las etapas  $k_i$ :

$$\mathbf{y}_{n+1}^{(8)} = \mathbf{y}_n + \sum_{i=1}^{12} b_i k_i \quad (3.24)$$

La estimación del error se obtiene comparando con las soluciones embebidas de orden 5 y 3:

$$e_{n+1}^{(5)} = \mathbf{y}_{n+1}^{(8)} - \mathbf{y}_{n+1}^{(5)} \quad (3.25)$$

$$e_{n+1}^{(3)} = \mathbf{y}_{n+1}^{(8)} - \mathbf{y}_{n+1}^{(3)} \quad (3.26)$$

Este doble control del error permite una mayor robustez y adaptabilidad del método.

### Características principales

- Orden de precisión: 8(5,3).
- Método explícito y adaptativo.
- Utiliza dos estimaciones embebidas para mayor robustez.

En `solve_ivp` (función de integración de EDO incluida en la paquetería SciPY, <https://scipy.org/>), si no se especifican manualmente, los valores predeterminados para las tolerancias son `rtol = 1 × 10-3` y `atol = 1 × 10-6`. Estas tolerancias controlan la precisión del método adaptativo: `rtol` limita el error relativo permitido entre pasos (como porcentaje del valor actual), mientras que `atol` establece un umbral mínimo absoluto por componente. En conjunto, definen la condición de aceptación del paso como:

$$\text{error} \leq \text{atol} + \text{rtol} \cdot |y|.$$

### 3.2.3. Métodos adaptativos implícitos y automáticos

Los métodos adaptativos implícitos y automáticos combinan estrategias de control de paso con esquemas de integración implícitos, siendo especialmente adecuados para sistemas rígidos. Estos métodos ajustan dinámicamente el tamaño del paso temporal y, en algunos casos, también el orden del método, con base en estimaciones del error local. A diferencia de los métodos explícitos, requieren resolver sistemas no lineales en cada paso, lo que permite una mayor estabilidad numérica, incluso con pasos grandes.

#### Método de Diferencias Backward (BDF)

[Gear, 1971] El método BDF es un método implícito de múltiples pasos, particularmente adecuado para sistemas rígidos. Es de orden variable, lo que significa que

utiliza mecanismos automáticos para cambiar el orden del método según la dinámica del sistema, es decir, adapta dinámicamente el número de pasos anteriores utilizados para optimizar la precisión y estabilidad en cada etapa de la integración. Su forma general es:

$$\sum_{j=0}^k \alpha_j \mathbf{y}_{n-j} = h \mathbf{f}(t_n, \mathbf{y}_n), \quad (3.27)$$

donde  $k$  es el orden del método y los coeficientes  $\alpha_j$  dependen de dicho orden.

### Características principales

- Orden de precisión: variable (generalmente entre 1 y 6).
- Método implícito.
- L-estable, adecuado para sistemas rígidos.
- Requiere resolver un sistema no lineal en cada paso.

El método BDF, al igual que DOPRI853, utiliza por defecto las tolerancias `rtol = 1e-3` y `atol = 1e-6` si no se especifican manualmente. En la implementación de `solve_ivp`, este método admite órdenes variables desde 1 hasta 5.

### Algoritmo LSODA

LSODA (Livermore Solver for Ordinary Differential Equations with Automatic method switching for stiff and non-stiff problems) [Hindmarsh, 1983] es un algoritmo adaptativo que combina dos estrategias:

- Método de Adams (explícito), para sistemas no rígidos.
- Método BDF (implícito), cuando detecta rigidez.

LSODA realiza un monitoreo dinámico del sistema y cambia automáticamente entre ambos esquemas según la rigidez detectada.

### Características principales

- Orden de precisión: variable.
- Método automático (explícito-implícito).
- Control adaptativo del error.
- Detecta y maneja la rigidez sin intervención del usuario.

### 3.3. Fuentes de error

Dado que existen múltiples factores que pueden falsear los resultados, queremos que los resultados obtenidos tengan una interpretación física apropiada. Sabemos que es casi imposible estimar con precisión el error. Sin embargo, hablaremos de aquellos que se consideran los más relevantes.

Los errores presentes en la integración numérica pueden clasificarse en:

- **Error de truncamiento local:** Error cometido en un solo paso, asociado a la aproximación de la derivada.
- **Error de truncamiento global:** Acumulación de los errores locales a lo largo de la integración.
- **Error de redondeo:** Error introducido por la representación finita de los números en la computadora.
- **Error de modelo:** Discrepancia entre el modelo matemático y el sistema físico real.

En este trabajo, el análisis se centra en el control y minimización de los errores de truncamiento y redondeo.

### 3.4. Control de error

El proceso de integración numérica de ecuaciones diferenciales ordinarias conlleva inevitablemente la introducción de errores, producto de las aproximaciones inherentes a los métodos utilizados. Estos errores pueden ser controlados y reducidos mediante técnicas adecuadas de estimación y ajuste dinámico del paso de integración. En esta sección se presentan los fundamentos del control de error [Butcher, 2016], distinguiendo entre los conceptos de error local y global, así como las estrategias implementadas para limitar su impacto en las simulaciones.

#### 3.4.1. Estimación de error local y global

##### Error local de truncamiento

El error local de truncamiento ( $e_{\text{local}}$ ) es el error cometido en un solo paso de integración, es decir, la diferencia entre la solución exacta y la solución numérica después de un único paso, asumiendo que el estado inicial es exacto. Este error es de orden  $\mathcal{O}(h^{p+1})$ , donde  $p$  es el orden del método utilizado y  $h$  el tamaño del paso.

### Error global acumulado

El error global ( $e_{\text{global}}$ ) es la acumulación de los errores locales a lo largo de la integración. Su magnitud depende tanto del orden del método como del tamaño del paso y del número total de pasos. En términos generales:

$$e_{\text{global}} = \mathcal{O}(h^p), \quad (3.28)$$

donde  $p$  es el orden del método. Para métodos adaptativos, el error global puede mantenerse controlado mediante el ajuste dinámico del paso, de modo que el error local en cada paso no exceda una tolerancia predefinida.

#### 3.4.2. Estrategias de control de paso

En los métodos de integración numérica, especialmente aquellos adaptativos, el control del tamaño del paso temporal es fundamental para equilibrar la precisión y la eficiencia computacional. Esta sección describe los mecanismos más comunes utilizados para aceptar, rechazar y actualizar el paso.

##### Métodos adaptativos

Los métodos adaptativos, como RKF45, DOPRI853, BDF y LSODA, incorporan un mecanismo de estimación del error local para ajustar dinámicamente el tamaño del paso de integración. Este proceso consiste en:

1. Calcular dos soluciones de distinto orden ( $\mathbf{y}_{n+1}^{(p)}$  y  $\mathbf{y}_{n+1}^{(p+1)}$ ).
2. Estimar el error local como la diferencia entre ambas soluciones:

$$\mathbf{e}_{n+1} = \mathbf{y}_{n+1}^{(p+1)} - \mathbf{y}_{n+1}^{(p)} \quad (3.29)$$

3. Comparar la norma del error con una tolerancia predefinida ( $\varepsilon$ ):

$$\|\mathbf{e}_{n+1}\| \leq \varepsilon \quad (3.30)$$

4. Ajustar el paso para el siguiente instante según:

$$h_{\text{nuevo}} = h \times \left( \frac{\varepsilon}{\|\mathbf{e}_{n+1}\|} \right)^{\frac{1}{p+1}} \quad (3.31)$$

Lo anterior, garantiza que el error local y global se mantenga dentro de un rango prestablecido.

##### Tolerancia y criterio de aceptación

En los métodos adaptativos, la decisión de aceptar o rechazar un paso depende de la comparación entre el error estimado y una tolerancia definida por el usuario. Se suele emplear un criterio mixto que involucra tolerancia relativa ( $\varepsilon_{\text{rel}}$  o  $\text{rtol}$ ) y absoluta ( $\varepsilon_{\text{abs}}$  o  $\text{atol}$ ):

$$\|\mathbf{e}_{n+1}\| \leq \varepsilon = \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}} \times \|\mathbf{y}_{n+1}\| \quad (3.32)$$

Si la condición no se cumple, el paso es rechazado y se repite con un tamaño reducido. Si se cumple, el paso es aceptado y el tamaño del siguiente paso se ajusta según la fórmula presentada anteriormente.

### Métodos no adaptativos

En los métodos no adaptativos (Euler, RK4), el paso  $h$  permanece constante durante toda la integración. La única forma de controlar el error consiste en seleccionar un paso suficientemente pequeño, lo que aumenta el costo computacional y puede amplificar el error de redondeo. Por lo que requiere la elección cuidadosa del paso de tiempo.

## 3.5. Estabilidad, convergencia y orden

El desempeño de los métodos numéricos empleados para la integración de ecuaciones diferenciales ordinarias depende de sus propiedades de precisión, estabilidad y convergencia, así como de la correcta implementación de mecanismos de control y reducción de error. En esta sección se analizan dichos conceptos aplicados a los métodos considerados en este trabajo.

### 3.5.1. Precisión y orden de los métodos

La precisión de un método numérico está asociada al orden de truncamiento del error local cometido en cada paso de integración. Formalmente, un método es de orden  $p$  si el error local de truncamiento es:

$$e_{\text{local}} = \mathcal{O}(h^{p+1}) \quad (3.33)$$

y el error global acumulado es:

$$e_{\text{global}} = \mathcal{O}(h^p) \quad (3.34)$$

En la siguiente tabla se resumen los órdenes de precisión de los métodos utilizados:

Método	Orden local	Orden global
Euler	2	1
RK4	5	4
RKF45	5	4
DOPRI853	9	8
BDF	1–6	Variable
LSODA	Variable	Variable

Tabla 3.1: Resumen del orden y precisión de los métodos numéricos utilizados

Métodos de mayor orden permiten alcanzar una mayor precisión con un tamaño de paso relativamente grande, aunque a costa de un mayor costo computacional por paso.

### 3.5.2. Estabilidad numérica

La estabilidad numérica de un método se refiere a su capacidad para controlar el crecimiento de errores de redondeo y truncamiento durante la integración.

Un método numérico es estable si pequeños errores numéricos no crecen exponencialmente durante la integración. En este sentido, un algoritmo se dice que es inestable, cuando los pequeños errores de redondeo que se producen en cada paso se propagan a través de cálculos posteriores con efecto creciente, llegándose a obtener un resultado que, aunque teóricamente debería ser próximo a la solución, en la práctica no tiene nada que ver con ella.

Se distinguen dos conceptos principales:

#### Estabilidad absoluta

Un método es absolutamente estable en un dominio  $D$  del plano complejo si, para la ecuación lineal test:

$$\frac{dy}{dt} = \lambda y, \quad \lambda \in \mathbb{C} \quad (3.35)$$

la solución numérica satisface:

$$|y_{n+1}| \leq |y_n| \quad \text{para } h\lambda \in D \quad (3.36)$$

Para métodos explícitos, como Euler y RK4, la región de estabilidad es limitada y depende del tamaño de paso  $h$ . En presencia de rigidez, estos métodos requieren pasos extremadamente pequeños para mantener la estabilidad. Mientras que, métodos implícitos como BDF y algoritmos automáticos como LSODA poseen estabilidad *A-estable* o *L-estable*, lo que los hace adecuados para sistemas rígidos. [Hairer and Wanner, 1996]

Recordemos que un método numérico se dice *A-estable* si su región de estabilidad incluye todo el semiplano complejo izquierdo, es decir, si permanece estable para cualquier paso  $h$  cuando la parte real de  $\lambda$  es negativa. Este tipo de estabilidad es deseable para resolver sistemas con soluciones que decaen rápidamente. Por otro lado, un método es *L-estable* si, además de ser A-estable, la función de estabilidad tiende a cero cuando  $h\lambda \rightarrow -\infty$ , lo que garantiza que los modos rígidos se amortigüen numéricamente. Esta propiedad es crucial para la integración robusta de sistemas altamente rígidos.

### Condición de estabilidad

En sistemas reales, la estabilidad práctica requiere que:  $h < h_{crítico}$ , donde  $h_{crítico}$  depende de las características del sistema dinámico y del método empleado. En métodos adaptativos, este control se realiza automáticamente ajustando el tamaño de paso.

#### 3.5.3. Convergencia

Un método numérico es convergente si la solución aproximada  $\mathbf{y}_n$  tiende a la solución exacta  $\mathbf{y}(t_n)$  cuando el tamaño de paso tiende a cero:

$$\lim_{h \rightarrow 0} \|\mathbf{y}_n - \mathbf{y}(t_n)\| = 0 \quad (3.37)$$

Bajo condiciones adecuadas de consistencia y estabilidad, los métodos numéricos considerados en este trabajo son convergentes.

### 3.6. Eficiencia computacional de los métodos numéricos

Además de la precisión y estabilidad, un aspecto fundamental en la evaluación de los métodos numéricos utilizados en este trabajo es su eficiencia computacional. La eficiencia de un integrador está relacionada con el tiempo requerido para completar la simulación y con la cantidad de operaciones necesarias para alcanzar un nivel de precisión específico.

Al implementar los métodos en un entorno computacional como Python, la eficiencia depende de diversos factores:

- **Número de evaluaciones de la función dinámica:** Cada evaluación de la función  $\mathbf{f}(t, \mathbf{y})$  representa un costo computacional. Métodos adaptativos y de mayor orden requieren más evaluaciones por paso.

- **Tamaño de paso:** Un paso pequeño implica mayor número de iteraciones, mientras que un paso grande puede comprometer la precisión y estabilidad.
- **Cálculo de derivadas o resolución de sistemas:** Los métodos implícitos, como BDF y LSODA (cuando detectan rigidez), requieren la resolución de sistemas de ecuaciones no lineales en cada paso, aumentando el costo computacional.
- **Control adaptativo de paso:** Los métodos adaptativos ajustan el paso dinámicamente, lo que puede implicar pasos rechazados y mayor cantidad de cálculos.
- **Lenguaje e implementación:** El uso de Python y librerías optimizadas como SciPy permite reducir el impacto computacional, aunque el lenguaje interpretado implica un costo mayor respecto a lenguajes compilados.

En particular, para el simulador se utilizó `solve_ivp` en Python, el cual ofrece una buena eficiencia computacional para sistemas de tamaño pequeño a medio. Si bien no alcanza la velocidad de integradores compilados como ODEPACK o CVODE, resulta suficientemente robusto y flexible para aplicaciones de simulación científica.

## 3.7. Comparación y análisis teórico de los métodos numéricos

La elección del método numérico para la simulación de la dinámica de vuelo depende de un balance entre precisión, estabilidad y costo computacional. En esta sección se realiza un análisis comparativo de los métodos implementados, considerando sus características teóricas y su desempeño práctico observado en las simulaciones realizadas.

### 3.7.1. Criterios de comparación

Para evaluar los métodos numéricos, se consideran los siguientes criterios:

- **Orden de precisión:** Relacionado con la velocidad de convergencia al disminuir el paso temporal.
- **Estabilidad numérica:** Capacidad de mantener el crecimiento de errores controlado.
- **Costo computacional:** Tiempo de cómputo requerido y complejidad algorítmica.
- **Adaptabilidad:** Capacidad de ajustar dinámicamente el paso de integración.
- **Eficiencia numérica:** Número de evaluaciones realizadas.

### 3.8. Comparación y validación mediante casos con solución analítica

Con el objetivo de validar la correcta implementación de los métodos numéricos y analizar su desempeño en situaciones controladas, se realizaron simulaciones de tres casos dinámicos simples cuya solución analítica se muestra en el Capítulo 2. Estos casos permiten evaluar cuantitativamente la precisión, estabilidad y eficiencia de los métodos considerados, así como identificar sus ventajas y limitaciones en función de la naturaleza del sistema.

Recordemos que los tres casos simples son los siguientes:

1. Caso 1: Tiro vertical con gravedad.
2. Caso 2: Tiro parabólico con arrastre lineal.
3. Caso 3: Tiro parabólico con arrastre cuadrático.

Las simulaciones se realizaron con seis valores diferentes del tamaño de paso:  $\Delta t = \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$  y un tiempo maximo  $t_{max} = 20s$ . En el caso de los métodos adaptativos, se definieron tolerancias absoluta de  $10^{-6}$  y relativa de  $10^{-9}$ .

#### 3.8.1. Condiciones iniciales y parámetros de simulación

Para los tres casos de prueba se emplearon las siguientes condiciones iniciales:

- Posición inicial:  $x_0 = 0, y_0 = 0$ .
- Velocidad inicial:  $v_{x0} = 50 \text{ m/s}, v_{y0} = 50 \text{ m/s}$ .
- Aceleración de la gravedad:  $g = 9.81 \text{ m/s}^2$ .
- Masa de la partícula:  $m = 1 \text{ kg}$ .
- Coeficiente de arrastre lineal y cuadrático:  $k = 0.1 \text{ kg/s}$ .

#### 3.8.2. Validez de la comparacion numerica entre los 40 casos evaluados

La comparación numérica realizada en esta sección se basa en un conjunto de 40 configuraciones diferentes que combinan métodos de integración numérica con distintos esquemas de control del paso de tiempo.

Se consideraron los siguientes tipos de métodos:

- Métodos no adaptativos: Euler y Runge-Kutta de orden 4 (RK4), evaluados con 6 valores de  $\Delta t$  fijos.

- Métodos adaptativos restringidos: RK45, BDF, LSODA y DOP853, evaluados con valores fijos de  $\Delta t$  usando la opción `max_step`.
- Métodos adaptativos libres: los mismos métodos adaptativos anteriores, ejecutados en modo automático sin restricción del paso.

Esta matriz de combinaciones cubre adecuadamente el espacio método × control de paso, permitiendo evaluar de manera integral el comportamiento, la precisión y la eficiencia computacional.

### Múltiples métricas de evaluación

Para evitar sesgos derivados del uso exclusivo de una sola métrica, se evaluó cada caso utilizando:

- Error cuadrático medio ( $L_2$ ) en posición y velocidad.
- Número total de pasos ejecutados.
- Tiempo de cómputo real (en segundos).
- Distribución de pasos locales  $\Delta t_{\text{local}}$ .

El análisis a partir de estas métricas proporciona una visión completa de las propiedades de los métodos.

### Elección de las variables representativas: posición vs velocidad

Aunque se evaluaron errores tanto en la posición como en la velocidad, se priorizó el análisis basado en la variable **posición** por las siguientes razones:

- La posición representa la trayectoria directa del objeto y está más alineada con los objetivos del simulador.
- Es físicamente más interpretable y se relaciona con eventos críticos como el apogeo o el impacto.
- Es más sensible a errores acumulativos, lo que la hace más adecuada para evaluar la precisión del método de integración.
- La velocidad puede parecer precisa incluso si la posición falla, ya que refleja un comportamiento más local al ser la primera derivada.

Por lo tanto, se consideran las gráficas y métricas basadas en la posición como las más representativas para comparar los métodos y sustentar conclusiones.

### 3.8.3. Caso 1: Tiro vertical con gravedad

Este caso representa el lanzamiento vertical de un cuerpo sometido únicamente a la aceleración gravitacional, sin considerar resistencia del aire ni fuerzas adicionales. La solución analítica del sistema se obtiene resolviendo las ecuaciones:

$$\frac{dv}{dt} = -g, \quad \frac{dz}{dt} = v, \quad z(0) = 0, \quad v(0) = v_0. \quad (3.38)$$

El procedimiento se encuentra desarrollado en el Capítulo 2. Se utilizó dicha solución como referencia para validar la precisión y eficiencia de los métodos numéricos implementados.

#### Análisis de propiedades

En la Figura 3.2 se puede observar que el error de los métodos BDF y LSODA parece estancarse o disminuir muy lentamente, manteniéndose relativamente constante alrededor de  $10^{-5}$  (para BDF) y  $10^{-11}$  (para LSODA), independientemente del  $dt$  en el rango mostrado.

También, podemos ver que RKF45 y DOPRI853 son más precisos que RK4 a medida que aumenta el paso de tiempo  $dt$ . Sin embargo, RK4 es un método más simple y alcanza errores comparables.

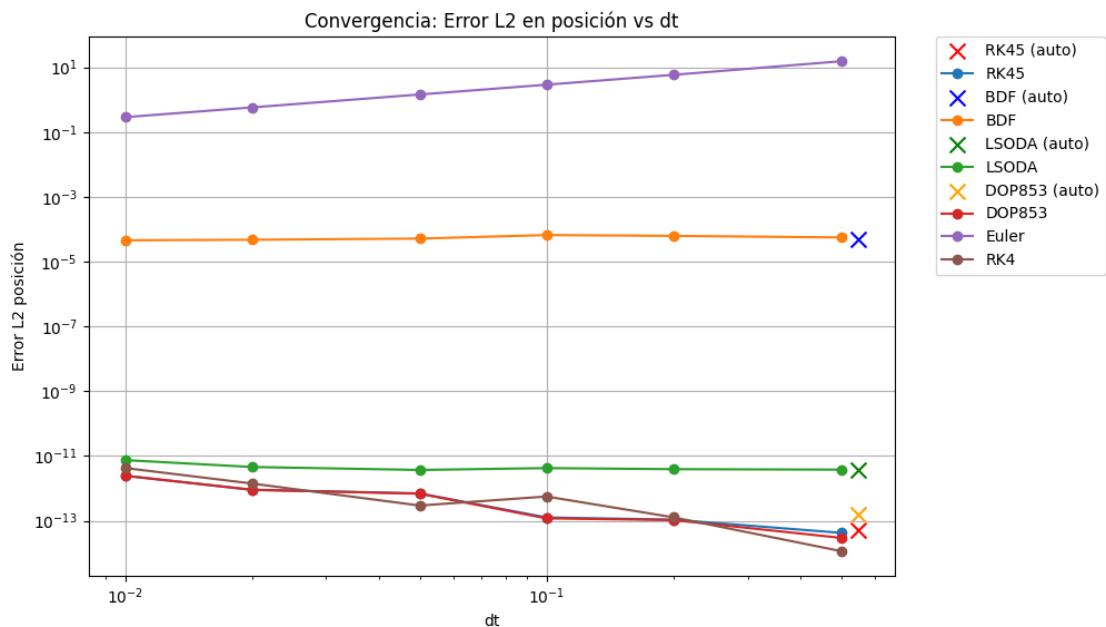


Figura 3.2: Convergencia de los métodos en el Caso 1

Por otro lado, en la Figura 3.3 se ilustra la eficiencia numérica de los métodos, resaltando con una estrella aquellos resultados con el ajuste de paso automático. De lo anterior, se observa que Euler resulta altamente ineficiente (como se esperaba por su bajo orden) y BDF logra un error de  $10^{-4}$  en un número considerable de pasos, lo que nos indica que el sistema no es rígido.

Así, RKF45 y DOPRI853, en particular las versiones automáticas, se sitúan en la esquina inferior izquierda indicando su bajo error del orden  $10^{-13}$  en aproximadamente 10 pasos. Mientras que, RK4 alcanza la precisión más alta pero requiere más de 3 veces más pasos para lograrlo.

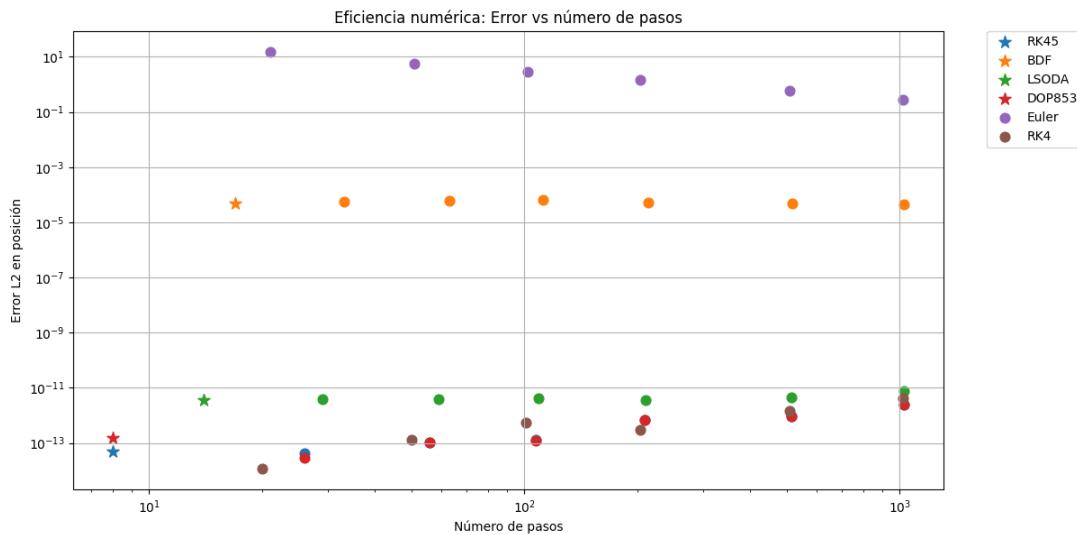


Figura 3.3: Eficiencia numérica de los métodos en el Caso 1

Finalmente, la Figura 3.4 muestra que los métodos de paso de tiempo fijo requieren un  $dt$  bajo, lo cual incrementa significativamente el tiempo de cómputo. Mientras que los métodos adaptivos destacan por obtener mejores resultados al permitir el ajuste automático que un ajuste limitado por un paso de tiempo máximo. Concluyendo que la eficiencia computacional de DOPRI853 es mayor. Notemos que aunque RK4 utiliza más pasos, su eficiencia computacional es mayor a la de RKF45 debido a que la complejidad de cada paso es menor al no tener que hacer cálculos extra del error.

Al comparar la altura de apogeo y el tiempo de apogeo estimados numéricamente contra el valor analítico se encontró que todos los métodos convergen al valor para pasos pequeños. Sin embargo, BDF y Euler presentan mayores desviaciones a medida que aumenta  $dt$ , lo que sugiere sensibilidad numérica.

De esta forma, evaluando del 1 al 5 las propiedades discutidas se obtiene la siguiente matriz de decisión que pondera criterios clave de los resultados en este Caso:

El método Dormand-Prince 8(5,3) (DOP853) obtuvo el mayor puntaje global, destacándose por su alta precisión, excelente estabilidad y control adaptativo eficiente. Le siguen de cerca los métodos RKF45 y LSODA, ambos con buen equilibrio entre

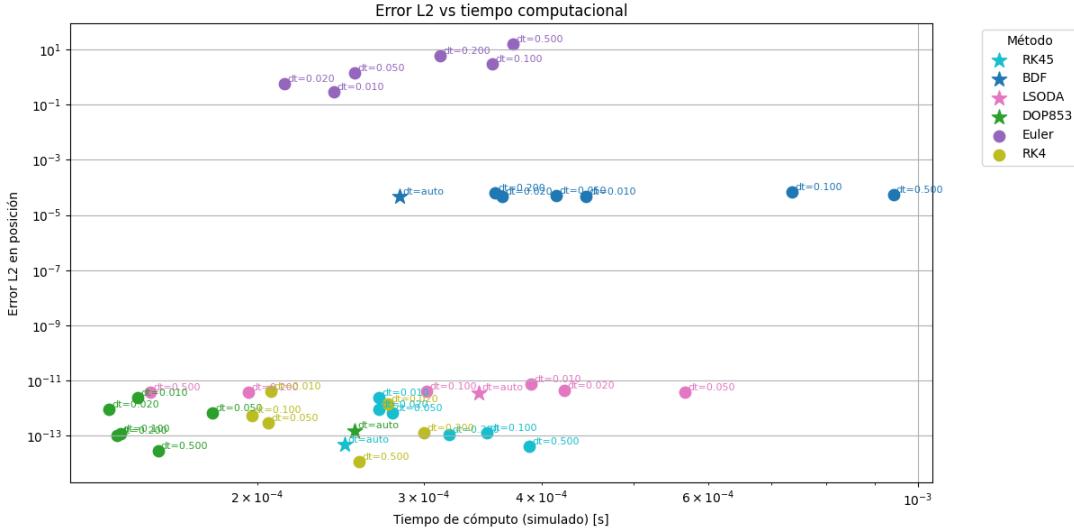


Figura 3.4: Eficiencia computacional de los métodos en el Caso 1

Tabla 3.2: Matriz de decisión Caso 1

Criterio	Peso	Euler	RK4	RKF45	DOP853	BDF	LSODA
Precisión (pos)	0.30	1	5	5	4	2	4
Eficiencia cómputo	0.20	3	4	3	5	2	4
Convergencia	0.15	1	3	5	5	2	4
Consistencia física	0.15	3	5	5	5	3	4
Estabilidad	0.20	2	5	5	5	4	5
<b>Total</b>		<b>1.90</b>	<b>4.50</b>	<b>4.60</b>	<b>4.70</b>	<b>2.55</b>	<b>4.20</b>

precisión y consistencia, aunque con mayor costo computacional. El método de Runge-Kutta de orden 4 (RK4) también mostró un buen desempeño. En contraste, el método de Euler fue descartado por su bajo rendimiento general, mientras que BDF mostró limitaciones en precisión y eficiencia para este tipo de problemas no rígidos. Este análisis respalda el uso de métodos de orden alto y control adaptativo como opciones preferentes para simulaciones de trayectoria que requieran alta fidelidad numérica.

### 3.8.4. Caso 2: Tiro parabólico con arrastre lineal

Este caso corresponde al lanzamiento de un proyectil en un plano vertical, considerando la influencia de la aceleración gravitacional y una fuerza de arrastre proporcional a la velocidad (modelo de arrastre lineal). No se contemplan otras fuerzas externas ni efectos adicionales. La solución analítica del sistema se obtiene resolviendo el conjunto de ecuaciones diferenciales y el desarrollo analítico detallado que se presenta en el Capítulo 2. Esta solución se utilizó como referencia para validar la precisión y eficiencia de los métodos numéricos.

ciencia de los métodos numéricos implementados, permitiendo evaluar su desempeño frente a un caso con solución exacta y dinámica bidimensional con dissipación.

### Análisis de propiedades

La gráfica de convergencia (Figura 3.5) muestra que la mayoría de los métodos implementados presentan una disminución del error al reducir el tamaño del paso de tiempo, lo cual confirma su comportamiento convergente. En particular, DOP853 con paso restringido (línea roja) se distingue por ofrecer la mayor precisión general, aunque evidencia una saturación de error para valores muy pequeños de  $dt$ , posiblemente por efectos de redondeo. Los métodos RK4 y RKF45 también muestran pendientes pronunciadas, reflejando su alto orden de convergencia. Asimismo, las versiones adaptativas ('auto'), muestran que LSODA(auto) es el más preciso, siendo el valor del error superior a los resultados obtenidos de usar  $dt_{max}$ .

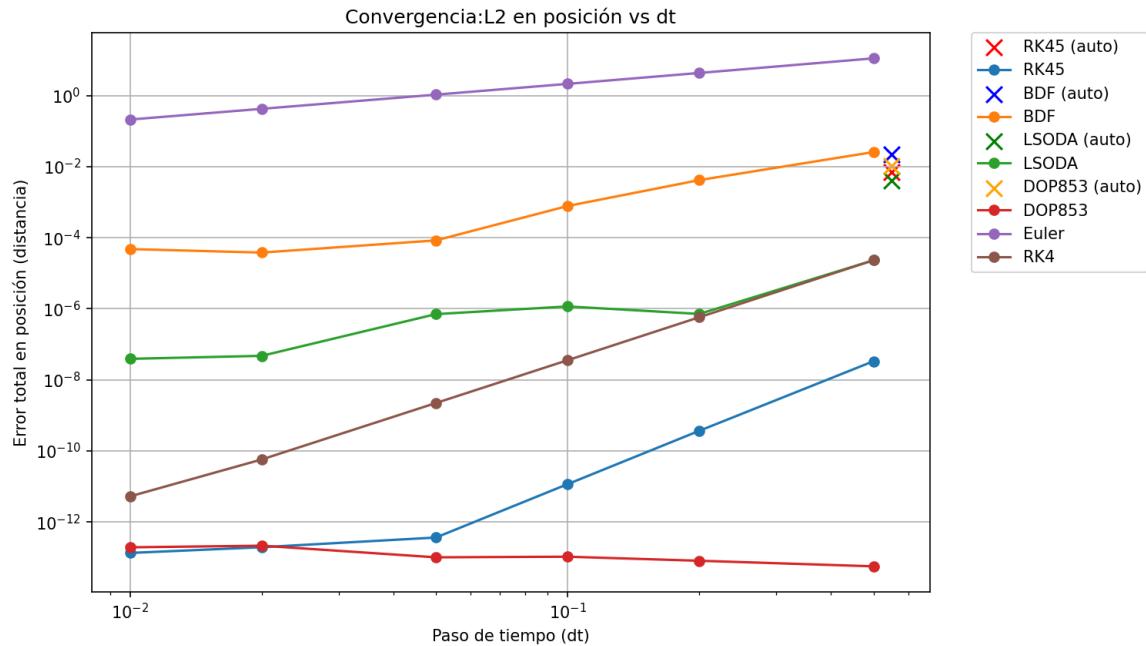


Figura 3.5: Convergencia de los métodos en el Caso 2

La gráfica de eficiencia numérica (Figura 3.6) muestra que, en general, los métodos más eficientes son aquellos que alcanzan un bajo error con un menor número de pasos. DOP853 y RK45 destacan como las opciones más precisas, especialmente en sus versiones con paso fijo, donde logran errores inferiores a  $10^{-12}$  utilizando entre 100 y 1000 pasos. Las versiones con paso automático (representadas por estrellas) requieren menos pasos que sus contrapartes de paso fijo, pero presentan errores mayores, lo cual refleja un compromiso entre precisión y eficiencia. En la mayoría de los métodos, se observa una tendencia clara: a mayor número de pasos, menor error, lo que confirma el comportamiento esperado de convergencia.

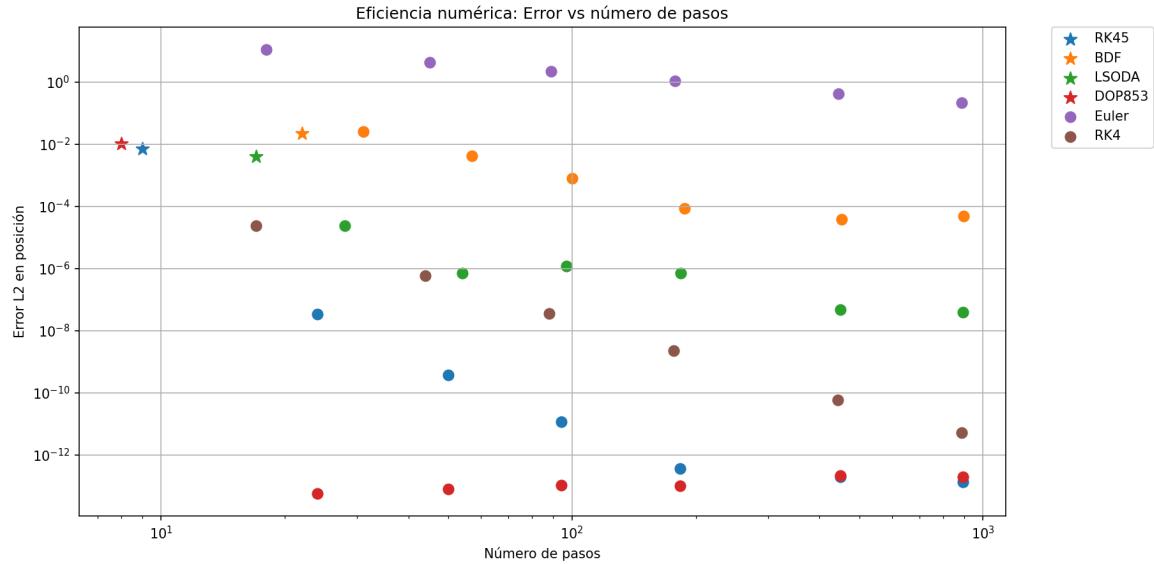


Figura 3.6: Eficiencia numérica en función del número de pasos para el Caso 2.

Para terminar el análisis de este caso, la gráfica del comportamiento adaptativo (Fig. 3.7) revela cómo cada método ajusta dinámicamente el tamaño del paso de tiempo  $dt$  a lo largo de la simulación. DOP853 y RK45 se muestran como los métodos más agresivos, incrementando rápidamente  $dt$  en las primeras etapas y alcanzando valores máximos cercanos a 7 y 6 segundos, respectivamente, lo que sugiere una estrategia orientada a la eficiencia.

En contraste, BDF adopta un enfoque más conservador, con incrementos escalonados y valores más moderados, reflejando su orientación hacia problemas con posible rigidez. LSODA presenta un comportamiento intermedio, con incrementos bruscos y tramos de  $dt$  constante, lo que puede estar asociado a su capacidad de alternar entre integradores rígidos y no rígidos según la dinámica del sistema. La siguiente matriz resume el desempeño observado para cada característica: Así, DOP853 destacó como el

Tabla 3.3: Matriz de decisión Caso 2

Criterio	Peso	Euler	RK4	RKF45	DOP853	BDF	LSODA
Precisión (pos)	0.30	1	5	5	5	2	4
Eficiencia comp	0.20	2	4	4	5	2	4
Convergencia num	0.15	1	5	5	5	2	4
Consist físca	0.25	2	4	5	5	3	4
Estabilidad gen	0.10	2	5	5	5	4	5
<b>Total ponderado</b>	1.00	<b>1.75</b>	<b>4.55</b>	<b>4.85</b>	<b>4.95</b>	<b>2.55</b>	<b>4.25</b>

mejor integrador en todos los frentes: precisión, eficiencia y fidelidad física. RKF45 se posicionó como una alternativa con gran estabilidad. Métodos implícitos como BDF y

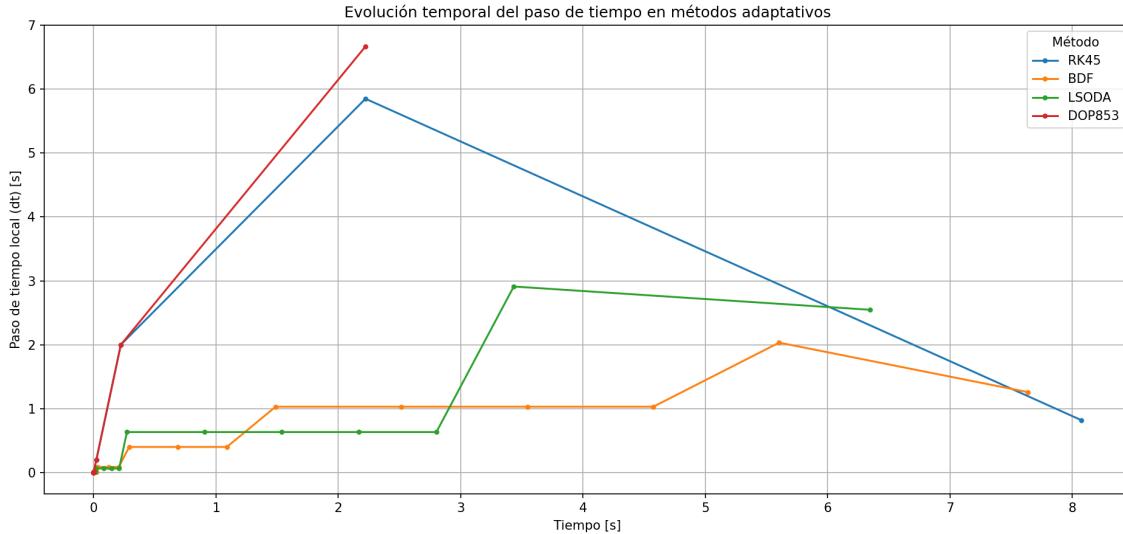


Figura 3.7: Evolucion temporal del paso de tiempo en el Caso 2

LSODA obtienen buenos resultados, pero con mayores tiempos de cómputo y menor exactitud.

Este análisis subraya la importancia de adaptar el método numérico al tipo de dinámica modelada. La elección del integrador adecuado debe considerar no sólo la precisión deseada, sino también el costo computacional, la rigidez del sistema y el grado de no linealidad.

### 3.8.5. Caso 3: Tiro parabólico con arrastre cuadrático

Este caso considera la fuerza de arrastre proporcional al cuadrado de la magnitud de la velocidad (modelo de arrastre cuadrático), representando un caso no lineal, con comportamiento más complejo que los anteriores.

Al igual que los casos 1 y 2, no se contemplan otras fuerzas externas ni efectos adicionales. La solución aproximada del sistema se obtiene resolviendo el conjunto de ecuaciones diferenciales y utilizando una expresión simplificada descrita en el Capítulo 2. Esta solución se utilizó como referencia para evaluar un sistema más cercano a la realidad.

#### Análisis de propiedades

La Figura 3.8 muestra el comportamiento del error  $L^2$  en la posición para distintos pasos de tiempo, donde no se observa una tendencia clara de disminución del error al reducir el paso de tiempo en los métodos de paso fijo, lo que sugiere que el paso de tiempo es demasiado grande o que el error ya es muy pequeño y está saturado numéricamente.

Por su parte, los métodos adaptativos con paso de tiempo restringido presentan fluctuaciones en el error, reflejando posibles dificultades para mantener una precisión estable bajo control manual del paso. Sin embargo, los métodos con control completamente adaptativo muestran consistentemente los errores más bajos a lo largo de la simulación. En particular, DOP853 en su versión adaptativa destaca por alcanzar el menor error global entre todos los métodos analizados, confirmando su robustez y eficiencia frente a dinámicas más complejas y no lineales.

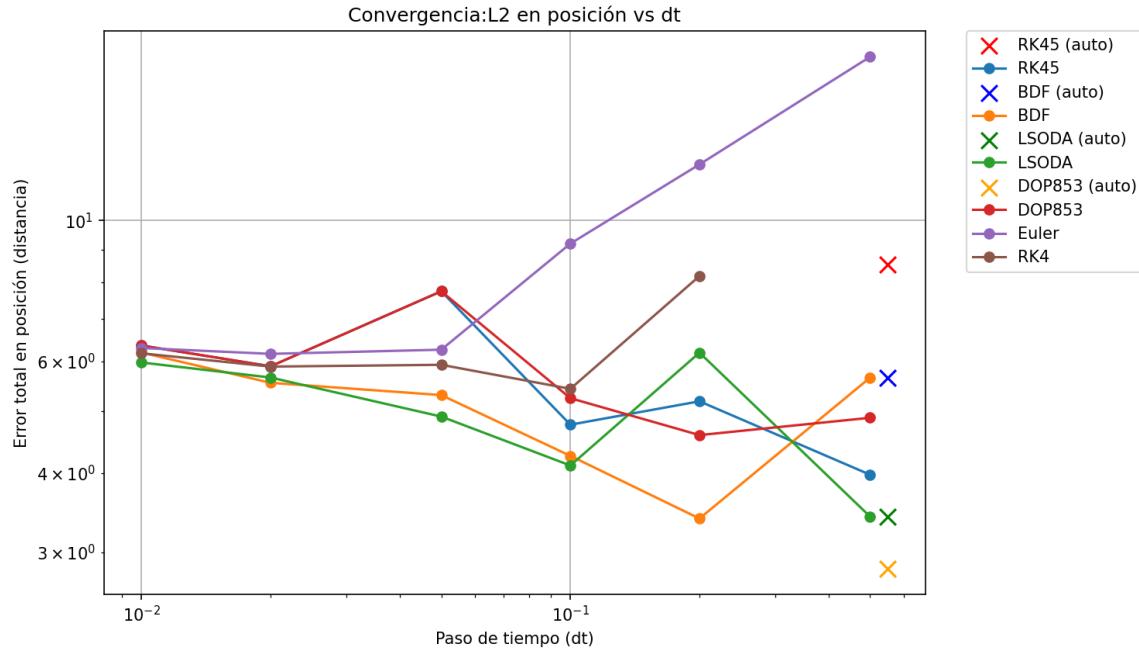


Figura 3.8: Convergencia de los métodos en el Caso 3

Por otro lado, la gráfica 3.9 analiza la eficiencia numérica y se observa que los métodos DOP853 y LSODA son los más eficientes, ya que logran alcanzar errores bajos utilizando un número relativamente bajo de pasos (entre 10 y 100 pasos). DOP853 tiene un punto con el error más bajo y muy pocos pasos (12). Notemos que conforme aumenta el número de pasos para todos los métodos, el error converge a un único valor.

Basándose en el tiempo de cómputo, los métodos adaptativos DOP853 y LSODA son superiores para este problema, ofreciendo la mejor combinación de alta precisión y rapidez computacional. RKF45(auto) es una alternativa eficiente también. De lo anterior, se concluye que los métodos de paso fijo, aunque pueden alcanzar buena precisión con  $dt$  pequeños, lo hacen a costa de un mayor tiempo de ejecución.

Para concluir este análisis se observa la evolución del paso de tiempo en los métodos adaptativos. De manera similar a lo que se observó en el caso previo, el comportamiento de la adaptación del paso de tiempo ( $dt$ ) a lo largo de la simulación en esta gráfica sigue patrones análogos. Nuevamente, se distingue la estrategia de RK45 y DOPRI853, que varían su  $dt$  de forma más agresiva alcanzando valores máximos más altos, frente a

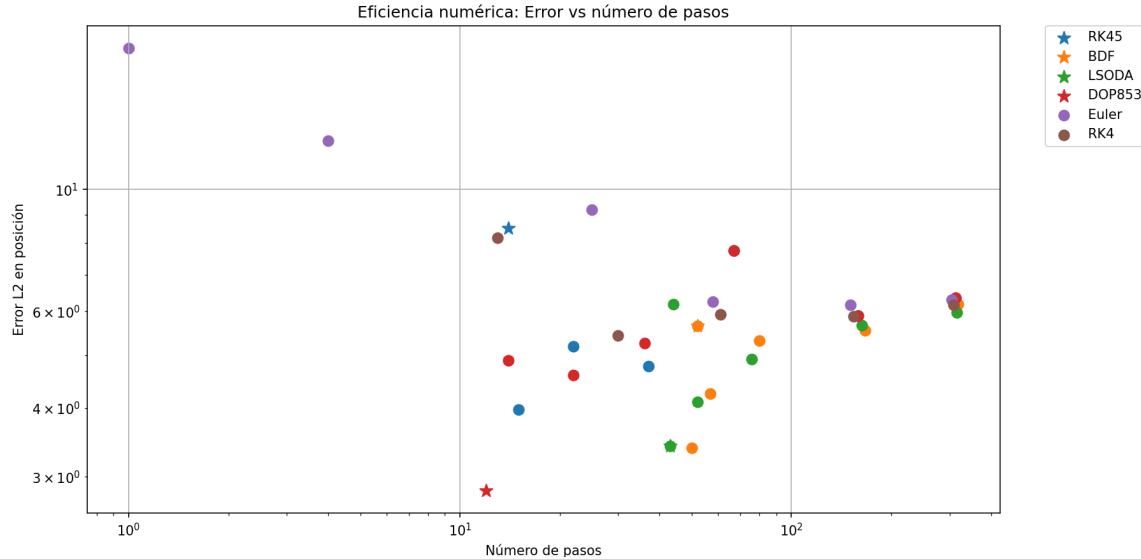


Figura 3.9: Eficiencia numérica en el Caso 3

la estrategia más conservadora de BDF y LSODA, que mantienen pasos de tiempo consistentemente más pequeños durante la integración.

Por lo que se obtiene la siguiente ponderación en la matriz de decisión:

Tabla 3.4: Matriz de decisión Caso 3

Criterio	Peso	Euler	RK4	RKF45	DOP853	BDF	LSODA
Precisión (pos)	0.30	1	5	5	5	2	4
Eficiencia comp	0.20	2	4	4	5	2	4
Convergencia num	0.15	1	5	5	5	2	4
Consist física	0.25	2	4	5	5	4	4
Estabilidad gen	0.10	2	5	5	5	4	5
<b>Total ponderado</b>	1.00	<b>1.75</b>	<b>4.55</b>	<b>4.85</b>	<b>4.95</b>	<b>2.55</b>	<b>4.25</b>

El análisis de este caso muestra que los métodos adaptativos son particularmente adecuados para dinámicas no lineales con términos cuadráticos como el arrastre aerodinámico. DOP853 y RKF45 se destacan tanto en precisión como en eficiencia.

LSODA puede ser una alternativa con configuración automática, mientras que BDF puede ser útil si se anticipan condiciones de rigidez. RK4, aunque efectivo en algunos contextos, pierde validez para pasos de integración grandes. Euler, nuevamente, se confirma como un método poco adecuado para dinámicas con no linealidades.

Este caso refuerza la importancia de utilizar integradores con control adaptativo del error en simulaciones aeroespaciales, donde la fidelidad numérica es crítica para la predicción del comportamiento dinámico del vehículo.

### 3.9. Discusión técnica y recomendaciones

Con el propósito de sintetizar las principales características de los métodos numéricos analizados en este capítulo, se presentan a continuación dos tablas resumen. La primera tabla (Tabla 3.5) contiene las propiedades técnicas relevantes de cada método, tales como orden de precisión, adaptabilidad, estabilidad, eficiencia y tipo de integrador. Esta información permite comparar objetivamente el desempeño numérico de los métodos.

Método	Orden	Adaptivo	Estabilidad	Eficiencia	Tipo
Euler	1	No	Condisional	Baja	Explícito
RK4	4	No	Condisional	Media	Explícito
RKF45	4(5)	Sí	Condisional	Alta	Explícito
DOPRI853	8(53)	Sí	Condisional	Muy alta	Explícito
BDF	1–6	Sí	A-estable / L-estable	Media	Implícito
LSODA	Variable	Sí (Auto)	A-estable / L-estable	Muy alta	Exp-Imp Auto

Tabla 3.5: Resumen de propiedades de los métodos numéricos utilizados

Por otra parte, la Tabla 3.6 resume la aplicación recomendada para cada método, estableciendo el contexto específico en el que su uso resulta más adecuado.

Método	Aplicación recomendada
Euler	Problemas simples y suaves, validación y referencia didáctica
RK4	Problemas no rígidos con dinámica suave y alta precisión
RKF45	Sistemas no rígidos con necesidad de control de error y ajuste dinámico de paso
DOPRI54	Sistemas no rígidos, simulaciones largas con control de error eficiente
BDF	Sistemas rígidos, problemas con alta sensibilidad y rigidez localizada
LSODA	Problemas generales con posibles fases rígidas y no rígidas; simulaciones robustas y eficientes

Tabla 3.6: Aplicación recomendada de los métodos numéricos utilizados

Los resultados obtenidos confirmaron que:

- Los métodos adaptativos (RKF45, DOPRI853, BDF, LSODA) mantuvieron el error global dentro de los límites especificados, ajustando dinámicamente el tamaño de paso.
- Los métodos no adaptativos (Euler, RK4) presentaron errores significativos para pasos grandes, siendo inestables en el Caso 3.

- En sistemas suaves (Casos 1 y 2), los métodos adaptativos explícitos (RKF45, DOPRI853) ofrecieron la mejor relación entre precisión y costo computacional.
- En el sistema rígido (Caso 3), los métodos implícitos y automáticos (BDF, LSODA) fueron los únicos capaces de mantener la estabilidad con pasos grandes.

### 3.10. Elección de integrador

La elección del integrador numérico para el simulador se realizó mediante una evaluación multicriterio que consideró cinco aspectos fundamentales: precisión en posición, eficiencia computacional, convergencia numérica, consistencia física y estabilidad general. Se compararon seis métodos: Euler, RK4, RKF45, DOP853, BDF y LSODA. Las matrices de decisión (Tabla 3.2, Tabla 3.3 y Tabla 3.4) mostraron que el método **Dormand-Prince 8(5,3) (DOP853)** obtuvo la puntuación más alta en los casos evaluados, consolidándose como el integrador preferente.

Este método se seleccionó como integrador principal para el simulador por las siguientes razones:

- Presentó la mayor precisión numérica, con errores globales mínimos incluso en simulaciones con pasos grandes.
- Su esquema adaptativo permite ajustar dinámicamente el paso de integración, optimizando la eficiencia sin comprometer la estabilidad.
- Logró el menor número de evaluaciones de la función dinámica en comparación con RKF45, reduciendo el tiempo total de simulación.
- Garantizó estabilidad en los escenarios probados sin necesidad de esquemas implícitos.

Adicionalmente, se recomienda el uso de **LSODA** como integrador alternativo en aquellos escenarios donde se sospeche la presencia de rigidez o donde se requiera un esquema automático robusto. Asimismo, el método **RKF45**, por su simplicidad y confiabilidad, podrá utilizarse en tareas de validación cruzada o simulaciones preliminares.

La selección adecuada del integrador constituye un elemento fundamental para garantizar la fidelidad numérica de las simulaciones, permitiendo representar con precisión el comportamiento dinámico del cohete en condiciones variadas y complejas.



## 4 Simulador de trayectoria

Como ya se comentó, la simulación y la modelación son herramientas fundamentales para la concepción y validación de cualquier producto, sistema o proceso. En este trabajo, se desarrolló un simulador de trayectoria para vehículos suborbitales considerando los elementos esenciales que permiten obtener respuestas coherentes con la física del vuelo, partiendo desde una formulación matemática rigurosa y adoptando un enfoque modular computacional.

El modelo se basa en representar únicamente las partes más importantes del sistema, reduciendo su complejidad sin perder precisión. Esta estrategia permite mantener una interpretación física lógica de los resultados y facilita la adaptabilidad del simulador a distintas condiciones. El código ha sido diseñado para ser sencillo de modificar, escalar o adaptar a diferentes escenarios y configuraciones, incluyendo:

- Diferentes ángulos de ataque
- Variaciones en la longitud del riel de lanzamiento
- Geometrías distintas (nariz, aletas, boattail, etc.)
- Condiciones atmosféricas cambiantes
- Ubicaciones geográficas de lanzamiento

Desde el punto de vista matemático, el **simulador resuelve un sistema de ecuaciones diferenciales ordinarias (EDO)** que modelan la dinámica del vuelo, considerando fuerzas gravitacionales, aerodinámicas y de propulsión. Además, se implementan condiciones iniciales y eventos discontinuos (como despliegue de paracaídas), lo que confiere al sistema un **carácter híbrido continuo-discreto**.

Se pone especial atención en la elección del método numérico utilizado para la integración, ya que este afecta directamente la estabilidad y precisión de los resultados. La formulación del modelo permite realizar análisis adicionales como:

- Estudios de convergencia: comparando soluciones a distintos pasos de tiempo.
- Estabilidad numérica: observando el comportamiento de las soluciones ante perturbaciones.

- Sensibilidad paramétrica: analizando cómo pequeñas variaciones en condiciones iniciales o parámetros del modelo afectan el resultado.

Estos aspectos refuerzan el enfoque teórico del simulador desde la perspectiva de la matemática aplicada.

## 4.1. Suposiciones

La modelación parte de una base simplificada que se complica progresivamente. Para construir un simulador práctico, se realizan las siguientes suposiciones, las cuales delimitan el espacio de validez del modelo:

- El cohete se modela como un cuerpo rígido con una distribución de masa previamente definida.
- La atmósfera se aproxima mediante el modelo ISA estándar (International Standard Atmosphere).
- El momento de inercia respecto al eje longitudinal es despreciado (no hay rotación axial).
- Las ecuaciones de movimiento son válidas únicamente para vuelo atmosférico; no se consideran efectos orbitales.
- Se ignoran efectos como la rotación terrestre o las fuerzas de Coriolis.

Estas hipótesis, aunque restrictivas, permiten un análisis matemático y computacional más accesible, manteniendo un equilibrio entre realismo físico y solvencia numérica. Además de que son suposiciones adecuadas para el régimen de vuelo que se está modelando.

## 4.2. Diseño del simulador

El simulador se estructura considerando tres entidades fundamentales:

1. Cohete: Incluye su geometría, masa, paracaídas y motor (ya sea híbrido o sólido).
2. Entorno de vuelo: Define las condiciones atmosféricas (temperatura, presión, densidad, velocidad del sonido) y el modelo de viento.
3. Condiciones iniciales del vuelo: Comprenden las características del riel de lanzamiento y la ubicación geográfica.

La Figura 4.1 ilustra el esquema general del simulador, detallando cómo interactúan estos componentes y cómo se integran en la solución numérica. En el núcleo

se encuentra la integración de las ecuaciones de movimiento, asistida por cálculos aerodinámicos, de fuerzas, momentos y propiedades físicas como CG y CP.

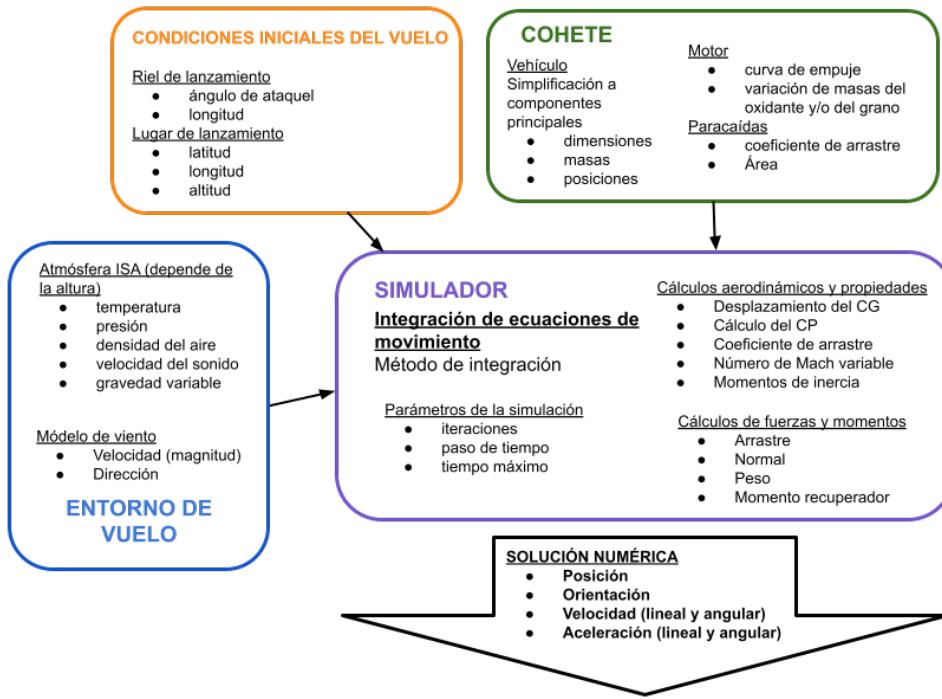


Figura 4.1: Esquema general de la simulación

Inicialmente, el simulador fue programado bajo un paradigma estructurado. Sin embargo, debido a la necesidad de manejar múltiples variables asociadas a cada componente, se migró a un enfoque orientado a objetos, lo que permitió una mayor claridad, escalabilidad y mantenimiento del código.

Esta estructura modular refleja una aproximación algebraica a sistemas dinámicos, en la cual cada clase actúa como un operador o conjunto de ecuaciones que define el comportamiento de una parte del sistema. Por ejemplo, los métodos que evalúan el centro de gravedad, presión o fuerzas aerodinámicas pueden entenderse como operadores funcionales aplicados sobre el estado del vehículo.

Desde una perspectiva matemática, este paso también permite encapsular ecuaciones, parámetros y condiciones dentro de clases, facilitando el manejo abstracto de operadores diferenciales y parámetros físicos como funciones miembro.

El desarrollo se realizó en Python utilizando las bibliotecas *numpy* y *scipy* para el cálculo, y *matplotlib* para la generación de gráficas. Cada entidad del sistema (cohete, motor, entorno, etc.) se encapsuló en una clase con atributos y métodos propios. Esto facilitó el acceso organizado a la información y permitió una arquitectura modular.

Esta estructura orientada a objetos facilita también el análisis matemático del sistema en términos de sistemas de ecuaciones acopladas. La posibilidad de integrar distintas clases y métodos en tiempo de ejecución permite considerar nuevas condiciones o

parámetros sin alterar el núcleo del modelo, lo que es congruente con el concepto matemático de sistemas paramétricos y la extensibilidad del espacio de soluciones.

El programa final está dividido en múltiples archivos interconectados, lo que permite una navegación y edición más sencilla. Esta estructura resulta especialmente útil al adaptar el simulador a diferentes vehículos y escenarios. Además, el paradigma orientado a objetos permite la reutilización del código para casos generalizados o para extender el modelo a cohetes multietapa, sistemas de control o simulaciones estocásticas.

Por lo anterior, el simulador fue organizado en varios paquetes y submódulos, con el objetivo de mantener la separación conceptual entre física, integración numérica, herramientas auxiliares, experimentación y visualización de resultados. Los paquetes principales y sus responsabilidades son:

- **PaqueteFisica/**: Contiene las clases principales relacionadas con el entorno físico del vuelo y el modelo del cohete.
- **PaqueteEDOs/**: Métodos numéricos para integración.
- **utils/**: Herramientas de visualización y transformaciones geométricas.
- **src/**: Ejecución principal de simulaciones reales, como `VueloLibre.py` y `VueloParacaidas.py`. Así, como la creación del vehículo a utilizar `XittleFile.py`.
- **CasosSimples/**: Pruebas de validación matemática en escenarios simplificados.
- **VarParams/**: Estudios de sensibilidad variando parámetros.
- **Visualizacion/**: Animaciones, gráficas y comparación de resultados.
- **GUI/**: Interfaz gráfica (en desarrollo) para uso práctico del simulador.

La base del simulador se encuentra en el repositorio **3DOF-Rocket-PU**, desarrollado por mí durante mi participación en el equipo de Propulsión UNAM y complementado para esta tesis. El objetivo principal es resolver numéricamente la trayectoria de un cohete suborbital modelado como un sistema de tres grados de libertad (3DOF), considerando la dinámica traslacional en el espacio tridimensional. El simulador implementa múltiples integradores numéricos para resolver sistemas de ecuaciones diferenciales ordinarias (EDOs):

- Euler
- RK4
- Runge-Kutta-Fehlberg 45 (RKF45)
- Dormand Prince 8(53)
- BDF- *Backward Differentiation Formula*
- Algoritmo LSODA

Estos métodos están contenidos en el paquete `PaqueteEDOs`, donde cada integrador está diseñado para recibir una función de derivadas que describe la dinámica del sistema, definida en el módulo `vuelo.py` dentro del paquete `PaqueteFisica`. Esta función de derivadas utiliza objetos físicos instanciados a partir de clases como `Cohete`, `Atmosfera`, `Viento`, y `Paracaídas` para calcular las fuerzas, momentos y aceleraciones que rigen el movimiento del vehículo. La clase `Vuelo` coordina estos cálculos y proporciona la interfaz principal entre el modelo físico y el método de integración.

El diseño del simulador se dividió en tres niveles funcionales principales:

1. **Nivel físico:** Incluye la implementación de modelos para el cohete, su motor, paracaídas, riel de lanzamiento, y entorno atmosférico. Estos módulos definen las ecuaciones físicas del sistema.
2. **Nivel numérico:** Implementa integradores para resolver el sistema de EDOs que describen la dinámica del cohete.
3. **Nivel de control y análisis:** Encargado de gestionar la simulación completa, establecer condiciones iniciales, coordinar eventos como el despliegue del paracaídas y salida del riel, y generar salidas como gráficas o animaciones.

Esta estructura permite realizar simulaciones de vuelo de manera precisa y controlada, integrando las ecuaciones de movimiento con condiciones realistas y propiedades dinámicas extraídas de pruebas experimentales. Además, ofrece flexibilidad para extender el simulador hacia nuevas condiciones, etapas múltiples, análisis estructurales o control activo.

### 4.3. Desarrollo del simulador

Desde el punto de vista numérico, se presta especial atención a la función que representa el sistema de ecuaciones diferenciales de primer orden. Esta función contiene las derivadas temporales de todas las variables de estado (posición, velocidad, orientación, velocidad angular, etc.), que se integran en el tiempo mediante distintos métodos.

La integración de las ecuaciones de movimiento se realiza mediante una función principal que evalúa las derivadas del sistema. La Figura 4.2 muestra el flujo lógico de dicha función `fun-derivs`, que corresponde a la implementación explícita de las ecuaciones diferenciales del sistema dinámico. El diseño de esta función se ajusta a la notación matemática clásica de problemas de valores iniciales:  $\frac{dy}{dt} = f(t, y(t))$

Además, el control del ciclo de simulación se realiza mediante un flujo principal de decisión que regula los eventos de vuelo como la salida del riel, el apogeo, el despliegue de paracaídas e impacto. Este flujo representa una lógica discreta de eventos que actúa sobre el sistema continuo. Matemáticamente, esto puede interpretarse como una implementación de un sistema dinámico híbrido, donde coexisten dinámicas continuas (las EDO) con eventos discontinuos (cambios en condiciones o fuerzas externas).

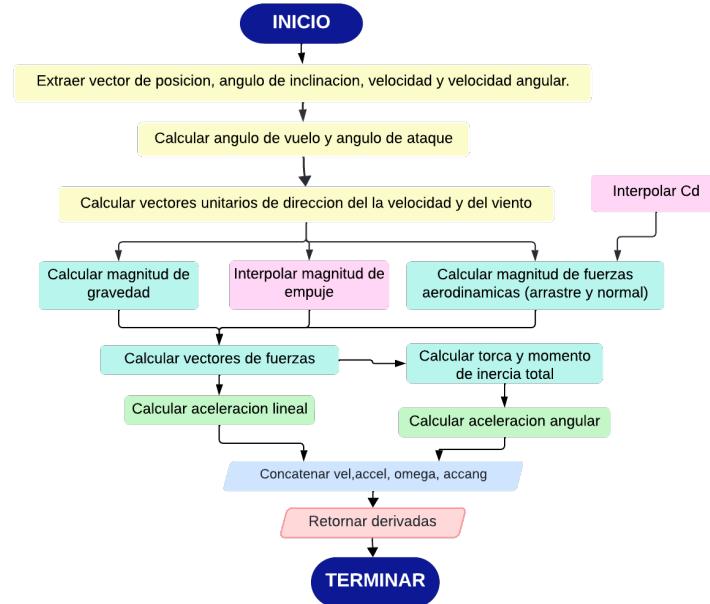


Figura 4.2: Diagrama de la función fun-derivs

Estos cálculos se realizan en iteraciones sucesivas para avanzar en el tiempo de forma discretizada, simulando así la evolución de la trayectoria del cohete. La Figura 4.3 presenta el diagrama completo de este flujo.

### 4.3.1. Clases utilizadas y sus métodos

La estructura del simulador se basa en una colección de clases que representan los elementos físicos, ambientales y numéricos del sistema. A continuación, se describen las clases principales y sus métodos más relevantes:

#### Viento2D-Viento3D

Esta clase genera un vector de viento bidimensional con magnitud y dirección aleatorias dentro de un rango determinado. Su constructor acepta como parámetros la velocidad media y la variación de la velocidad. Los métodos principales permiten obtener la representación del viento como vector (de 2 o 3 dimensiones) y su visualización. Esta clase es útil para simular condiciones realistas y dinámicas del entorno atmosférico.

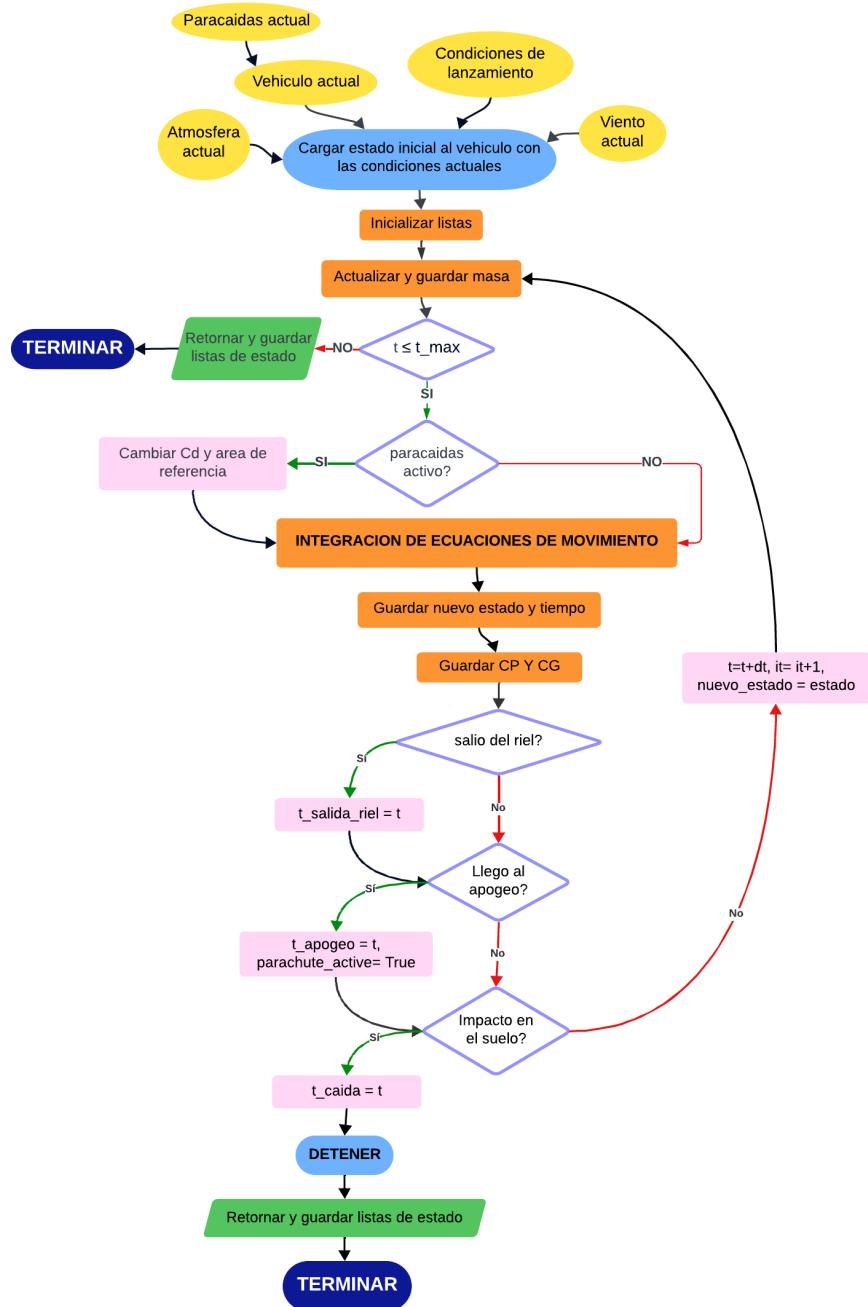


Figura 4.3: Diagrama de flujo de simular vuelo

### Torre de lanzamiento

Representa las condiciones iniciales del riel de lanzamiento, incluyendo la longitud y el ángulo de elevación. Es una clase sencilla pero crucial para definir la orientación y posición inicial del vehículo.

## Atmósfera ISA

Modela la atmósfera estándar (ISA) hasta 80 km de altitud. Incluye:

- `altitudGeopotencial()`: Calcula la altitud geopotencial en función de la altura geométrica.
- `determinarCapa()`: Determina la capa atmosférica correspondiente a una altitud dada.
- `calcPropiedades()`: Devuelve presión, densidad, temperatura y velocidad del sonido en función de la altura.

Esta clase implementa fórmulas físicas basadas en la termodinámica atmosférica y es fundamental para evaluar fuerzas dependientes de la altitud.

Para el futuro se piensa incluir las siguientes capas de la atmósfera y poder simular un vuelo de hasta 100 km de apogeo.

## Integradores

Cada método de integración (Euler, RK2, RK4, RKF45, etc.) está encapsulado en una clase que recibe como entrada una función de derivadas del sistema. Esta estructura permite cambiar de integrador sin modificar el sistema físico. Se considera buena práctica computacional y permite aplicar criterios de estabilidad y control del error.

El constructor inicializa el atributo `funderivadas` con la función proporcionada por el usuario. Esta función debe tener la forma  $f(t, state)$ , donde  $t$  es el tiempo y  $state$  es el vector de estado.

## Componentes

Para este trabajo, el cohete está representado por el conjunto de sus componentes principales. Se modela cada parte del cohete como una masa con forma geométrica simple (cono, cilindro, etc.) de acuerdo a la Tabla 4.1, y son posicionados sobre el eje de simetría del cohete, cuyo origen está en la punta de la nariz, siguiendo el *sistema de coordenadas SP*.

De esta forma, se asignan atributos como masa, longitud, dimensiones y posición sobre el eje transversal. En cada clase se realiza el cálculo del centro de gravedad, centro de presión y momento de inercia individual. Se definió una superclase `Componente` de la cual heredan `Cono`, `Cilindro`, `Aleta`, etc., facilitando extensibilidad y consistencia.

<b>Componente</b>	<b>Forma</b>
Nariz	Cono
Aviónica	Cilindro
Carga útil	Cilindro
Paracaídas	Cilindro
Fuselaje	Cilindro
Aletas	-
Tanque de oxidante	Cilindro
Cámara de combustión	Cilindro
Grano de combustible	Cilindro
Boattail	-

Tabla 4.1: Figura estándar que representa a cada componente

## Cohete

Esta clase modela el cuerpo completo del vehículo como una composición de componentes geométricos individuales —como cilindros, conos, aletas y boattail— cada uno con propiedades físicas independientes: masa, dimensiones, posición, forma y coeficientes aerodinámicos. La clase calcula propiedades globales del vehículo como el centro de gravedad (CG), centro de presión (CP), momento de inercia y coeficientes de sustentación y arrastre.

Uno de los aspectos fundamentales es el manejo del empuje del motor, el cual se obtiene a partir de curvas experimentales. Estas curvas son interpoladas y su integral define el impulso total disponible. El consumo de propelente y la variación de la masa total también se modelan dinámicamente a partir de archivos de datos.

Adicionalmente, se incorpora el efecto del número de Mach sobre el coeficiente de arrastre, actualizando dicho coeficiente a cada paso temporal según la altitud y velocidad del vehículo. También se incluyen correcciones aerodinámicas basadas en el método de Barrowman, adaptado para condiciones supersónicas, lo que permite una representación más precisa del comportamiento del cohete a lo largo de su trayectoria atmosférica.

## Paracaídas

Este módulo representa los sistemas de recuperación del cohete. Define el coeficiente de arrastre y el área efectiva del paracaídas una vez desplegado. La clase se comunica con el objeto `Cohete` para modificar su aerodinámica y masa efectiva tras el despliegue. Su activación está determinada por eventos programados durante la simulación, típicamente después del apogeo o tras cierto retardo.

## Vuelo

Contiene la lógica principal para resolver las ecuaciones diferenciales del sistema. Dentro de esta clase se define la función `fun-derivs`, que computa las derivadas de posición, velocidad, aceleración y momento angular. Utiliza como entradas las condiciones actuales de estado y hace uso de las propiedades del cohete, la atmósfera y el viento. Calcula fuerzas como empuje, gravedad, arrastre y sustentación, y permite una formulación extensible para distintos tipos de vuelo (con y sin paracaídas, en vacío, etc.).

## A simular

La función `simular-vuelo` o los archivos `VueloLibre.py` o `VueloParacaidas.py` inicializan todos los objetos necesarios, establecen las condiciones iniciales de lanzamiento y el integrador a utilizar, y ejecutan la integración temporal hasta el impacto. Durante este proceso se monitorean eventos como la salida del riel, el apogeo y el despliegue de paracaídas. Los resultados pueden visualizarse con herramientas de animación o compararse contra datos experimentales para propósitos de validación.

Como resultado se obtuvo una herramienta funcional que vincula conceptos de dinámica, modelado, análisis numérico y validación experimental en una plataforma coherente y extensible.

## 4.4. Implementación del simulador

El simulador es adaptable a cualquier vehículo suborbital que cuente con los siguientes archivos de entrada en formato .csv:

1. Curva de empuje (empuje vs tiempo)
2. Variación de masa (masa vs tiempo)
3. Coeficiente de arrastre vs número de Mach

Parámetros adicionales definidos por el usuario incluyen:

- Tiempo máximo de simulación  $t_{max}$
- Paso de integración  $dt$

El código fuente del simulador desarrollado en este trabajo se encuentra disponible públicamente en el repositorio de GitHub: <https://github.com/Meijix/RocketTrajectorySimulator-PropulsionUnam>. Este repositorio contiene la implementación completa del modelo dinámico, los integradores numéricos, el generador de viento, la interfaz gráfica y los archivos de prueba utilizados durante el desarrollo del proyecto. Su publicación busca fomentar la transparencia, facilitar su reutilización por

otros equipos de cohetería experimental, y servir como base para futuras mejoras o adaptaciones en proyectos de simulación aeroespacial.

## 4.5. Interfaz gráfica

Como parte del trabajo realizado, se desarrolló un software completo para la simulación de trayectorias de vuelo de cohetes suborbitales, el cual incluye una interfaz gráfica de usuario (GUI) diseñada para facilitar la interacción con el modelo, sin requerir conocimientos avanzados en programación. Esta interfaz permite configurar el vehículo, definir las condiciones iniciales del vuelo, seleccionar el modelo de viento y ejecutar simulaciones numéricas con solo unos cuantos clics.

En la Figura 4.4 se muestra el módulo de definición del cohete, donde es posible especificar dimensiones, masas y geometría de cada componente del vehículo. Esta sección permite construir el modelo a partir de nariz, fuselaje, motor, aletas y boattail, con campos editables y botones para guardar o cargar configuraciones prediseñadas.

The screenshot shows a software interface for defining a rocket's geometry and components. The top navigation bar includes tabs for Cohete, Importar Curvas, Parámetros de Simulación, Trayectoria 3D, Posición/Velocidad, Fuerzas, Ángulos, Estabilidad, Viento Relativo, and Resumen. The Cohete tab is selected. The main area contains several sections with input fields:

- Nariz (Cono):**
  - Longitud (m): 0.81
  - Diámetro base (m): 0.152
  - Masa (kg): 0.8
  - Geometría: ogiva
- Aletas:**
  - Número de aletas: 4
  - Envergadura (m): 0.11
  - Cuerda raíz (m): 0.3
  - Cuerda punta (m): 0.1
  - Ángulo barrido (°): 25
  - Masa total aletas (kg): 0.0
- Fuselaje (Cilindro):**
  - Longitud (m): 3.336
  - Diámetro exterior (m): 0.152
  - Espesor pared (m): 0.003
  - Masa (kg): 0.0
- Boattail:**
  - Longitud (m): 0.12
  - Diámetro frontal (m): 0.152
  - Diámetro trasero (m): 0.132
  - Masa (kg): 0.000
- Motor:**
  - Masa (kg, sin prop.): 0.0
  - Longitud (m): 0.573
  - Diámetro (m): 0.152

At the bottom are three buttons: Actualizar Cohete, Guardar Definición, and Cargar Definición.

Figura 4.4: Módulo de definición del cohete en la interfaz gráfica.

En la Figura 4.5, se observa el módulo de importación de curvas, donde se cargan archivos de empuje, variación de masa en función del tiempo, así como variación de coeficiente de arrastre en función al número de Mach, esenciales para modelar la propulsión y las fuerzas aerodinámicas. Tras la importación, el sistema permite visualizar gráficamente las curvas cargadas, como se muestra en la gráfica inferior de dicha figura.

La Figura 4.6 muestra el panel de configuración de parámetros de simulación. Aquí se

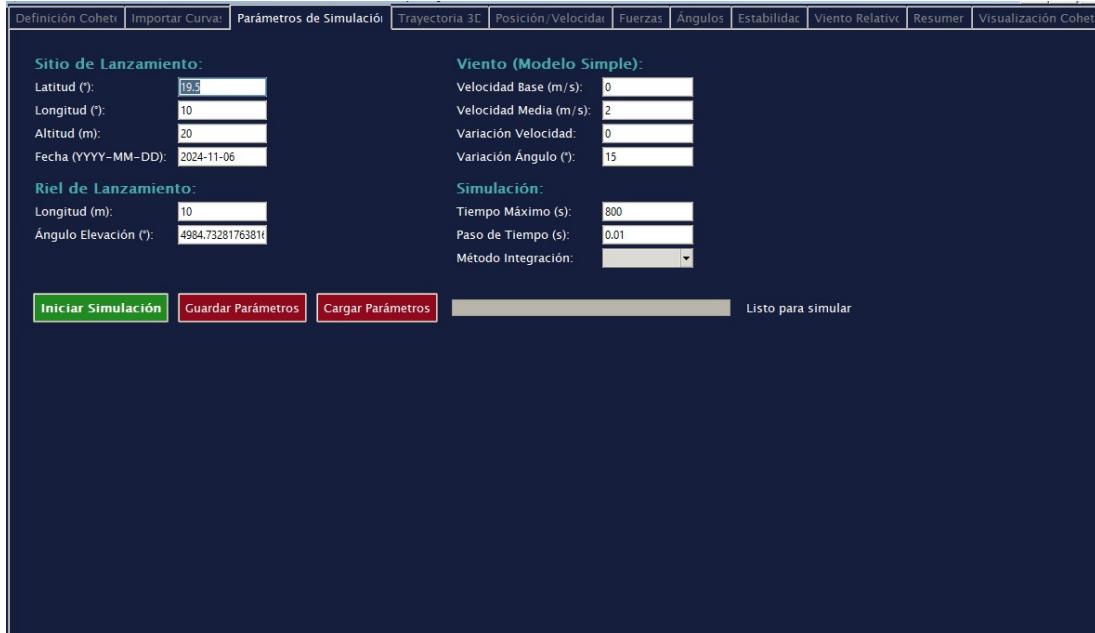


Figura 4.5: Importación y visualización de curvas de empuje y masa.

definen el sitio y riel de lanzamiento, las características del viento, el paso de integración, el tiempo máximo de simulación y el método numérico deseado. Esta flexibilidad permite realizar análisis de sensibilidad y adaptar las simulaciones a condiciones específicas de prueba.

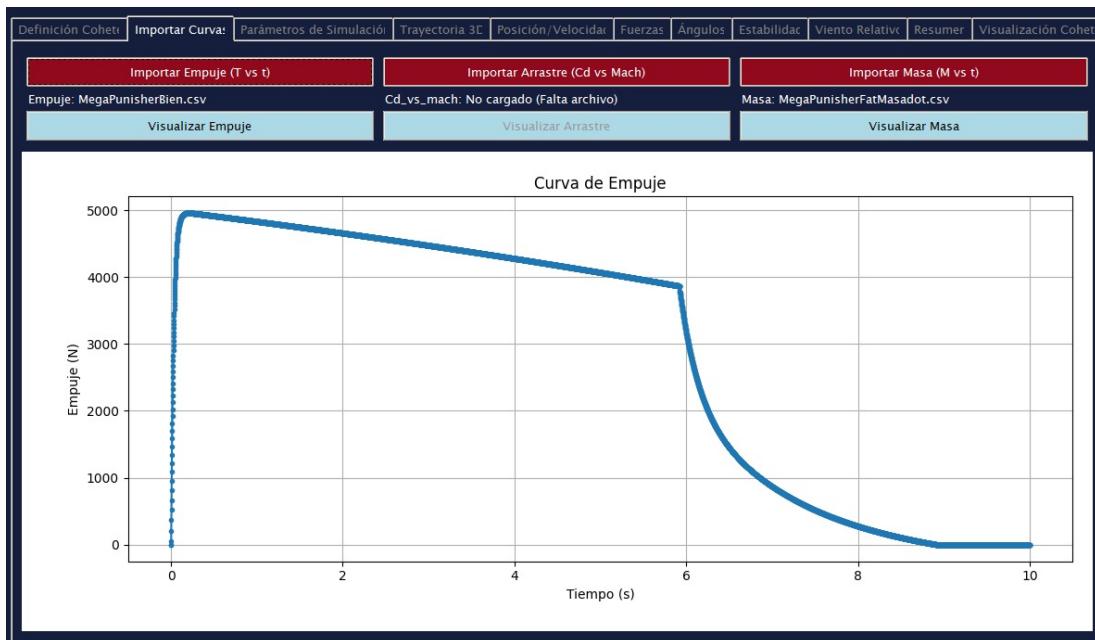


Figura 4.6: Parámetros de simulación: entorno, viento, riel y método de integración.

Inicialmente, la interfaz fue concebida bajo un modelo simplificado del cohete que permitía definir la geometría y masa del vehículo a través de tres elementos básicos:

nariz, fuselaje y aletas. Sin embargo, en fases posteriores del desarrollo, se implementó un modelo estructural más completo, donde el cohete se construye a partir de múltiples componentes internos y externos, tales como tanques, secciones estructurales, motores, sistemas de recuperación, entre otros. Esta evolución en la representación del vehículo implica una reformulación de la forma en que se ingresan y gestionan los parámetros del diseño, motivo por el cual la interfaz aún se encuentra en fase de actualización y no está completamente integrada con el nuevo esquema de entrada.

A pesar de ello, la versión actual de la interfaz ya permite modificar de manera intuitiva una gran variedad de parámetros clave, como dimensiones, masas, condiciones de lanzamiento (latitud, altitud, ángulo de elevación), características del viento, paso de integración, y duración máxima del vuelo simulado. Asimismo, el sistema permite guardar configuraciones para futuras simulaciones.

Esta flexibilidad convierte al simulador en una herramienta eficaz para iterar sobre el diseño del vehículo, evaluar la sensibilidad a distintas condiciones iniciales y validar decisiones estructurales y operativas. Una vez concluida la implementación completa del nuevo modelo de componentes, se espera que la interfaz permita definir detalladamente la configuración del cohete y realizar simulaciones con un nivel de realismo aún mayor, facilitando la optimización integral del diseño.

#### 4.5.1. Aplicaciones educativas y uso en equipos estudiantiles

Además de su valor como herramienta técnica para el diseño y simulación de cohetes, la interfaz gráfica desarrollada tiene un gran potencial como recurso educativo en cursos de dinámica de vuelo, simulación numérica o introducción a la ingeniería aeroespacial. Su diseño visual, modular e interactivo permite a estudiantes explorar cómo varían las trayectorias ante diferentes configuraciones geométricas, condiciones atmosféricas y parámetros de propulsión, fomentando así una comprensión más intuitiva de los fenómenos involucrados.

Esta herramienta resulta especialmente útil para equipos estudiantiles de cohetería experimental. La posibilidad de modificar parámetros y visualizar los efectos de manera inmediata permite iterar rápidamente en el diseño de vehículos, analizar distintos escenarios de lanzamiento y prever posibles fallas o condiciones críticas antes de llevar a cabo una prueba real.

El hecho de que el simulador utilice un modelo físico completo, basado en integración numérica de un sistema dinámico tridimensional, lo convierte en un punto medio entre software comercial (como OpenRocket) y herramientas de simulación avanzada desarrolladas en investigación. Esto proporciona a los estudiantes una plataforma accesible pero con fundamentos rigurosos, útil tanto en el aprendizaje como en el diseño profesional.

## 4.6. Análisis estático y dinámico: seguridad del vuelo

El análisis de estabilidad de un vehículo suborbital puede abordarse desde dos perspectivas complementarias: el análisis estático y el análisis dinámico. Ambos son fundamentales para garantizar que el vehículo mantenga su integridad y control durante el vuelo, especialmente en condiciones reales con perturbaciones externas.

### 4.6.1. Análisis estático

El análisis estático permite evaluar la estabilidad del cohete a partir de su configuración geométrica y distribución de masas, sin necesidad de simular su comportamiento temporal.

Este análisis es esencial en las etapas de diseño, y permite descartar configuraciones inherentemente inestables antes de realizar simulaciones completas o lanzamientos experimentales.

### 4.6.2. Análisis dinámico

El análisis dinámico evalúa el comportamiento del vehículo en el tiempo, resolviendo su dinámica lineal y rotacional bajo la acción de fuerzas y momentos variables. En este enfoque se consideran:

- Fuerzas de empuje, gravedad, arrastre y normal aerodinámica.
- Momentos estabilizadores y amortiguadores.
- Efectos del viento, paracaídas y consumo de masa.

La simulación dinámica permite observar si el vehículo:

- Mantiene su orientación nominal a lo largo del vuelo.
- Presenta oscilaciones angulares crecientes o amortiguadas.
- Regresa a su trayectoria ante perturbaciones externas.
- Llega al apogeo y desciende sin descontrol.

### 4.6.3. Relación con la seguridad del lanzamiento

Ambos análisis son indispensables para certificar la seguridad de un lanzamiento suborbital. Un diseño que no cumple con los criterios estáticos puede derivar en inestabilidad inmediata tras el despegue. Por otro lado, una configuración estáticamente estable puede fallar dinámicamente si presenta oscilaciones no amortiguadas o es muy sensible al viento.

En este trabajo, el simulador implementado permite realizar ambos tipos de análisis:

- Calculando el calibre en función del tiempo a partir de la evolución del  $CG$ .
- Simulando el comportamiento angular y espacial del cohete completo bajo distintas condiciones.

## 4.7. Limitaciones de la simulación

Aunque el simulador es robusto y modular, presenta ciertas limitaciones importantes:

- El vehículo se simplifica a 7-9 componentes principales. Aunque se pueden agregar más, esta es la configuración recomendada por eficiencia.
- No se consideran deformaciones estructurales del cohete ni efectos térmicos.
- No se realizan simulaciones de Monte Carlo (variación estadística de parámetros).
- No se evalúan cargas estructurales ni se realiza análisis de esfuerzos.
- El sistema no incluye aún control activo ni perturbaciones externas aleatorias.

Desde un enfoque matemático, estas limitaciones corresponden a restricciones sobre el modelo (formulación) más que sobre el método numérico. Sin embargo, el esquema actual puede servir de base para la extensión del modelo a sistemas de ecuaciones diferenciales parciales (EDP), sistemas de control, o incluso técnicas de reducción de modelos (model order reduction).

A pesar de estas simplificaciones, el simulador es una herramienta eficaz para la evaluación preliminar de trayectorias y para comparar distintos diseños o escenarios de misión. Su valor reside en su equilibrio entre simplicidad y rigor, permitiendo aplicar fundamentos matemáticos a problemas reales de ingeniería aeroespacial.



# 5 Validación y verificación: Caso real “Xitle II”

Dado que un vuelo real presenta muchas condiciones variables como las condiciones climáticas o el lugar de lanzamiento, entre otras, resulta un fenómeno complicado de simular.

En este capítulo se presenta la validación y verificación del modelo matemático y del simulador desarrollado, utilizando como caso de estudio el cohete experimental "Xitle II", desarrollado por el equipo de Propulsión UNAM. Se emplean datos obtenidos de pruebas reales de lanzamiento y herramientas de simulación comerciales.

Para la aceptación del software se retoman las definiciones propuestas por Roache [Roache, 1998] en el contexto de simulaciones computacionales, distinguiendo claramente entre verificación y validación:

**Verificación:** Es el proceso mediante el cual se evalúa si las ecuaciones del modelo han sido resueltas correctamente desde el punto de vista numérico. Esto implica estimar la incertidumbre numérica, causada por la elección del método de integración, el tamaño del paso de tiempo, y otros factores computacionales.

**Validación:** Es el proceso de determinar si el modelo matemático representa de manera adecuada el sistema físico real. En este caso, se estima la incertidumbre del modelado, evaluando la capacidad del modelo para reproducir datos experimentales bajo condiciones reales.

## 5.1. Caso práctico: Xitle 2

El cohete Xitle II, vehículo estándar para todo el trabajo, es un lanzador de propulsión híbrida, diseñado y manufacturado por el equipo Propulsión UNAM para competir en la Spaceport America Cup 2024, en la categoría de 30,000 ft (9,144 m) de sistemas tipo *SRAD* (*Student Researched and Developed*). Este vehículo es una iteración del proyecto original Xitle, el cual participó en la edición 2023 de la misma competencia.

El Xitle II tiene una longitud total de 4.266 metros, alcanza velocidades supersónicas y está impulsado por el motor híbrido *El Castigador*. Su estructura está construida



Figura 5.1: Fotografías del cohete Xitle 2 en Spaceport America Cup 2024

con fibra de carbono y fibra de vidrio laminada a mano.



Figura 5.2: CAD seccionado del cohete Xitle

Las dimensiones y pesos de cada sección están resumidos en la Tabla 5.1. De la misma

Componente	Masa [kg]	Longitud [cm]	Posición [m]
Nariz	0.8	81.0	0.0
Coples	1.5	17.6	81.0
Tubo recuperación	2.3	92.0	98.6
Transferidor de carga	1.0	25.0	190.6
Tanque vacío	8.7	125.0	215.6
Válvulas	2.4	16.7	340.6
Cámara de combustión	4.3	57.3	357.3
Boattail	0.251	12.0	414.6
<b>TOTAL</b>	<b>40</b>	<b>4.26</b>	<b>-</b>

Tabla 5.1: Tabla de masas y dimensiones de los componentes principales del cohete Xitle.

forma, los componentes internos se resumen en la Tabla 5.2. El sistema de recuperación del cohete Xitle II se encuentra alojado dentro de la bahía de recuperación, ubicada entre el cono de nariz y la bahía de aviónica, y está compuesto por dos paracaídas: el drogue y el principal, junto con los mecanismos de activación.

La separación entre el cono y el cuerpo del cohete está asegurada por *shear pins* (pernos de corte), los cuales se rompen al activarse el mecanismo de CO<sub>2</sub> en el apogeo,

Componente	Masa [kg]	Longitud [cm]	Posición [m]
Aviónica	1.8	21.0	20.0
Carga útil	4.3	30.0	50.0
Drogue	0.6	17.0	100.0
Main	1.7	30.0	140.0
Oxidante	12.0	133.0	215.6
Grano	4.0	50.5	357.3
Aletas	1.1	-	414.6
TOTAL	25.5	-	-

Tabla 5.2: Tabla de masas y dimensiones de los componentes internos del cohete Xitle.

permitiendo así la eyección del cono de nariz y el despliegue del paracaídas de arrastre. Este primer despliegue estabiliza el descenso hasta alcanzar aproximadamente 400 metros de altitud, momento en el que se activa un segundo mecanismo, el Tender Descender, que libera el paracaídas principal para reducir la velocidad final de caída a 8 m/s y asegurar una recuperación íntegra del vehículo.

Detalles más específicos del proyecto pueden encontrarse en los reportes técnicos de Propulsión UNAM para Spaceport America Cup 2023 y 2024 [UNAM, 2024]. Así como en los trabajos de tesis de Isaac Aztatzi Rubio [Rubio, 2023], Omar Córdoba Carbajal [Carbajal, 2024] y Sebastián Polo Pérez [Pérez, 2024].

Sin embargo, para nuestro propósito, es necesario conocer a grandes rasgos algunas características del motor y el sistema de adquisición de datos.

### 5.1.1. Datos del motor

El **motor híbrido Castigador V2**, que utiliza óxido nitroso ( $N_2O$ ) como oxidante y parafina sólida como combustible, alcanzó un **impulso total de 21,138.8713 N·s**, lo que lo ubica dentro de la categoría de motores tipo “O”. Con un tiempo de combustión (en diseño) de **14.24 segundos** y un **empuje promedio cercano a 1731 N**. Mientras que el empuje máximo registrado fue de 2389.8141 N, logrando un **impulso específico (ISP) de 143.02 segundos**, indicador clave de eficiencia en motores de propulsión.

La curva de empuje obtenida en una prueba de motor realizada el **28 de mayo del 2024** se muestra en la Fig. 5.3.

### 5.1.2. Adquisición de datos

Para la validación del simulador de vuelo desarrollado en este trabajo, fue fundamental contar con registros experimentales del comportamiento del vehículo durante

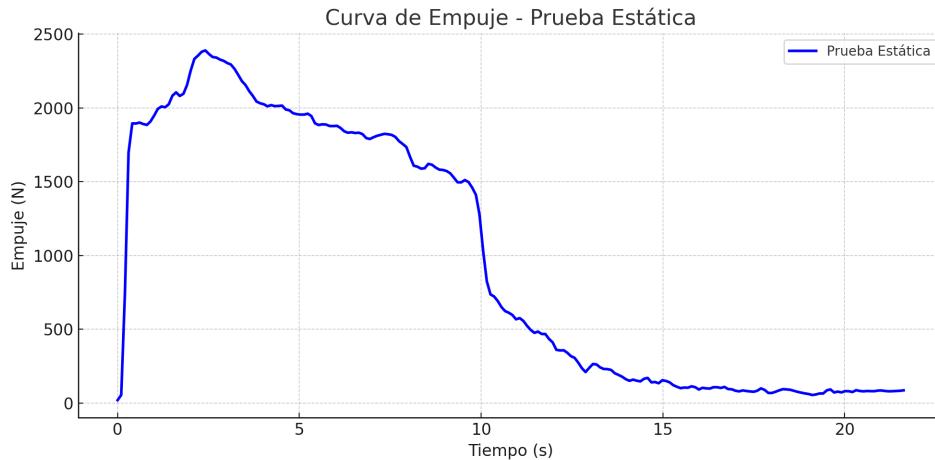


Figura 5.3: Curva de empuje- Prueba estática 28 de mayo, 2024

cada fase del vuelo. Con este objetivo, la **división de aviónica del equipo Propulsión UNAM** diseñó e integró varios módulos electrónicos encargados de adquirir y almacenar datos.

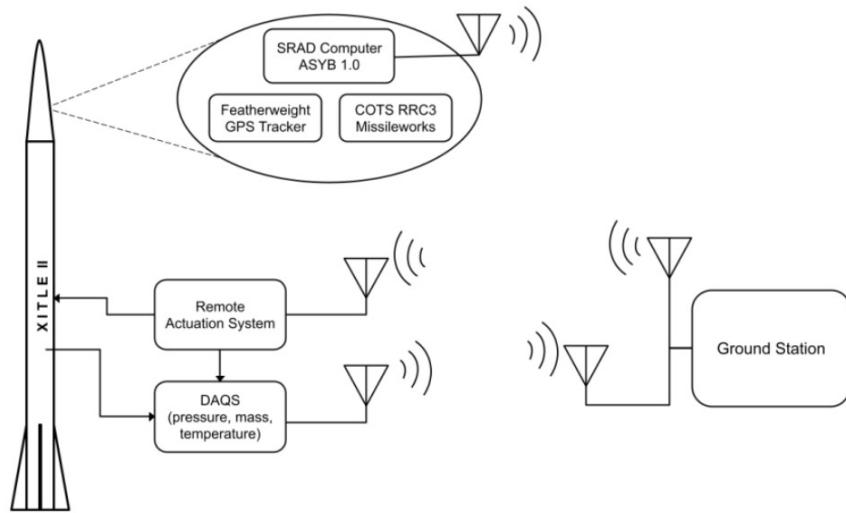
Estos sistemas fueron pensados para ofrecer redundancia, diversidad de fuentes y distintas modalidades de transmisión y almacenamiento, lo cual permitió reconstruir la trayectoria del cohete con suficiente fidelidad para comparar contra los resultados del modelo numérico.

A pesar de que el vehículo contaba con 3 distintos módulos que participaron directamente en la adquisición de datos de vuelo: la computadora de vuelo ASYB, la computadora comercial COTS RRC3 y el GPS Featherweight, los datos de comparación experimental se tomaron únicamente del GPS. A continuación, se describen brevemente los módulos y las razones de dicha decisión.

### Computadora de vuelo ASYB

La computadora de vuelo **ASYB**, cuyo nombre surge de manera informal como acrónimo de “Ahora Sí Ya Bien”, fue desarrollada durante la temporada 2023-2024 y es lo que se conoce como *"Student Researched and Developed"* o *SRAD* por sus siglas en inglés. Está basada en una Raspberry Pi Pico y adquiere datos mediante el sensor GY-87, que integra acelerómetro (MPU6050), giroscopio, magnetómetro (HMC5883L) y barómetro (BMP180).

Hay que mencionar que parte del segmento inicial del vuelo se perdió por un error humano en el cambio manual de frecuencia, razón por la cual no se cuenta con un perfil de vuelo completo que nos sirva de comparación.



**Fig. 23 General Overview of the interconnection between electronics system**

Figura 5.4: Interconexion General de los sistemas electronicos

### Computadora comercial COTS RRC3

La computadora de vuelo **RRC3**, de tipo comercial, fue utilizada como sistema redundante para el despliegue del sistema de recuperación. Cuenta con un sensor barométrico de alta resolución y registra continuamente la altitud durante el vuelo.

Almacena los datos localmente y no transmite en tiempo real. Por lo anterior, y dado que el vehículo resultó dañado, no se lograron recuperar dichos datos.

### GPS Featherweight

El **Featherweight GPS Tracker**, encargado de nuestros datos experimentales, es un sistema autónomo especializado en vuelos de cohetes que proporciona datos completos de posición geográfica.

Las variables registradas son:

- coordenadas geográficas de la estación terrena
- coordenadas geográficas del cohete
- Altitud sobre el suelo del cohete
- velocidad horizontal y vertical
- tiempo GPS
- head (dirección en grados)
- Intensidad de señales entre el cohete y la estación terrena

- Voltajes de las baterías
- Distancia horizontal y total entre la estación terrena y el cohete

Este dispositivo transmite en tiempo real por radiofrecuencia y también almacena datos localmente, generando un archivo CSV y KML.

## 5.2. Modelado del vehículo

Los valores estándar para estas simulaciones son los correspondientes al vehículo **Xitle II**. Obteniendo como resultados las siguientes características:

Componente	Masa [kg]	Long[cm]	CG [cm]	CP [cm]	CN $\alpha$	$I_x$ [kg·m <sup>2</sup> ]
Nariz	0.8	81.0	0.6075	0.3775	2.0	0.0043
Aviónica	1.8	21.0	10.5	0.0	0.0	0.0031
Carga útil	4.3	30.0	15.0	0.0	0.0	0.0079
Coples	1.5	17.6	8.8	0.0	0.0	0.0106
Tubo recuperador	2.3	92.0	46.0	0.0	0.0	0.0341
Transf de carga	1.0	25.0	12.5	0.0	0.0	0.0093
Tanque vacío	8.7	125.0	62.5	0.0	0.0	0.0424
Válvulas	2.4	16.7	8.35	0.0	0.0	0.0062
Cámara combustible	4.3	57.3	28.65	0.0	0.0	0.0227
Boattail	0.251	12.0	3.95	4.37	1.2762	0.0011
Drogue	0.6	17.0	8.5	0.0	0.0	0.0010
Main	1.7	30.0	15.0	0.0	0.0	0.0031
Oxidante	12.0	133.0	66.5	0.0	0.0	0.0380
Grano	4.0	50.5	25.25	0.0	0.0	0.0074
Aletas	1.1	-	0.0	17.06	2.7683	0.0043
<b>COHETE</b>	<b>46.75</b>	<b>426.6</b>	<b>252.35</b>	<b>313.42</b>	<b>7.42</b>	<b>0.0628</b>

Tabla 5.3: Propiedades geométricas y aerodinámicas de los componentes del cohete Xitle.

Nótese que el cálculo del centro de gravedad, centro de presión, masas del vehículo y momentos de inercia de los componentes permanecen constantes. En la siguiente figura (Fig. 5.5) se puede observar la localización de los centros de gravedad y presión generales y de algunos componentes.

## 5.3. Vuelo Experimental

El vuelo experimental del que se adquirieron datos se realizó el 22 de junio de 2024 durante la competencia Spaceport America Cup 2024.

- **Sitio de lanzamiento:** Spaceport America, Las Cruces, Nuevo México

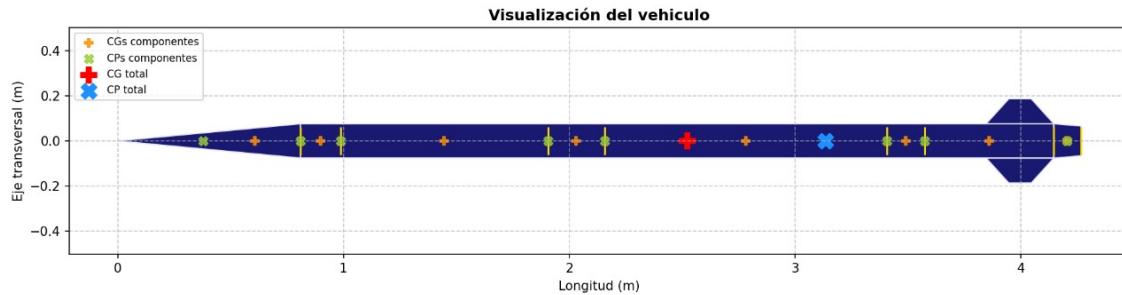


Figura 5.5: Visualización del Xitle 2

- **Coordenadas:**  $32^{\circ} 59' 25''$  N,  $106^{\circ} 58' 11''$  O
- **Altitud:** 1,401 m sobre el nivel del mar
- **Velocidad del viento:** 4-5 m/s
- **Dirección del viento:**  $230^{\circ}$
- **Longitud del riel:** 10 m
- **Inclinación del riel:**  $88^{\circ}$

### 5.3.1. Datos del GPS

En esta sección se muestran los resultados obtenidos por el GPS Featherweight durante el vuelo experimental. Este dispositivo permitió registrar en tiempo real la altitud y las componentes vertical y horizontal de la velocidad, proporcionando así una visión clara del perfil dinámico seguido por el vehículo.

La Figura 5.6 muestra la altitud registrada en función del tiempo. El vehículo alcanza un apogeo de **6573.36 m** a los **50 s** tras el despegue. Este punto representa el máximo alcanzado durante el vuelo, a partir del cual se inicia la fase de descenso.

En la Figura 5.7 se presentan las velocidades horizontal y vertical. La **velocidad vertical** alcanza un valor máximo de **925.00 m/s** alrededor de los **21.4 s**, mientras que la **velocidad horizontal** alcanza un pico de **255.00 m/s** aproximadamente a los **26.4 s**. Estos valores reflejan un perfil de vuelo con fuerte aceleración inicial, seguido de un deslizamiento horizontal continuo durante el ascenso.

Ambos registros son consistentes con una trayectoria balística suborbital, en la que el cohete sigue un arco ascendente impulsado inicialmente por el motor y posteriormente por la inercia, antes de iniciar la caída libre.

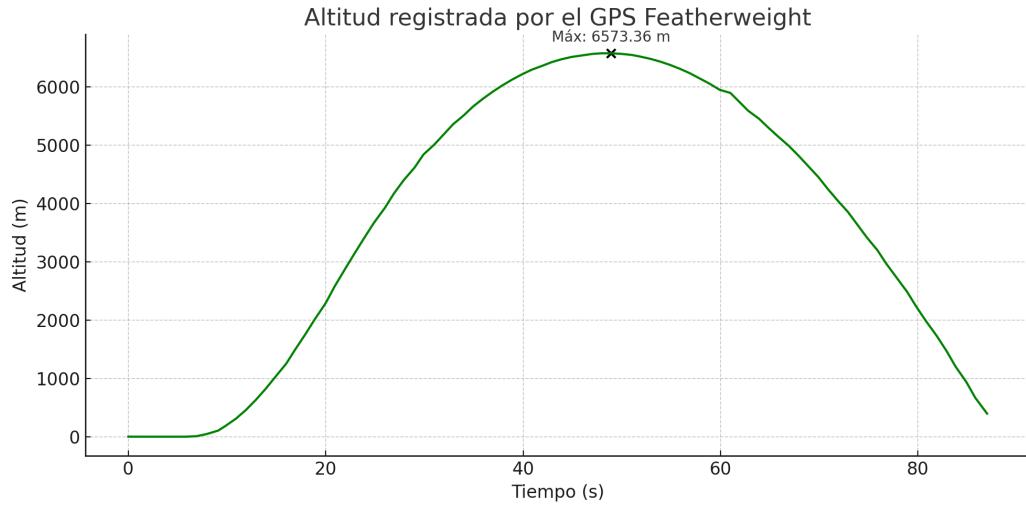


Figura 5.6: Altitud registrada por el GPS Featherweight.

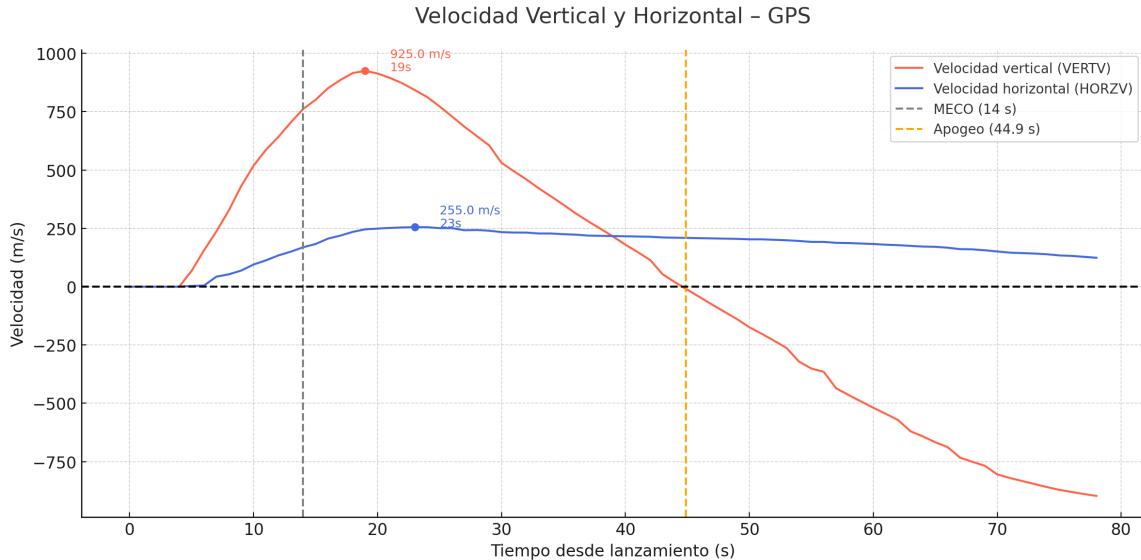


Figura 5.7: Velocidades vertical y horizontal registradas por el GPS Featherweight.

## 5.4. Vuelo Openrocket

En esta sección, se muestran los resultados de la simulación generada en OpenRocket con los mismos parámetros de entrada, permitiendo una comparación directa de los resultados.

El modelo representativo del vehículo en el software OpenRocket es el siguiente:

La Figura 5.9 muestra la evolución de la altitud del cohete durante la simulación realizada en OpenRocket. Se observa una trayectoria ascendente suave durante la fase propulsada, seguida de una parábola balística que culmina en una **altitud máxima**

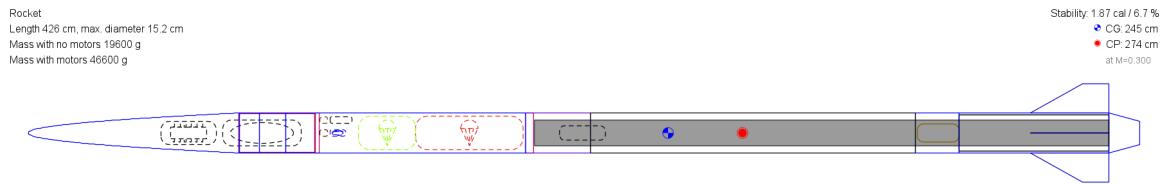


Figura 5.8: Esquema del cohete Xitle en el programa Open Rocket

de **5885.7 m** alcanzada aproximadamente a los **38 segundos** de vuelo. Este valor se encuentra por debajo de los datos experimentales, lo cual será discutido más adelante en la sección de comparación.

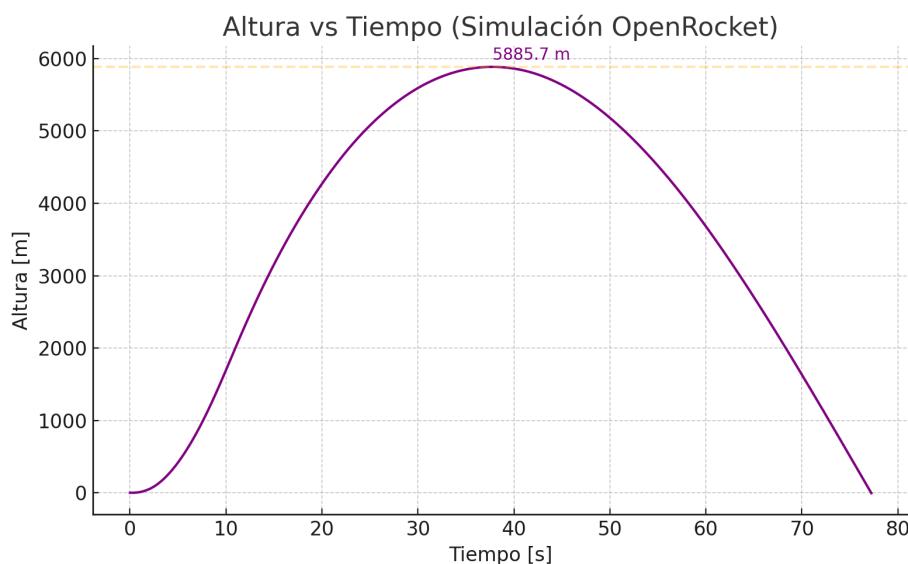


Figura 5.9: Altura del cohete en función del tiempo durante la simulación en OpenRocket

La Figura 5.10 representa las velocidades vertical y lateral del vehículo a lo largo del tiempo. La velocidad vertical muestra un incremento brusco durante el ascenso motorizado, alcanzando una **velocidad vertical máxima de 315.66 m/s**, seguida de una caída continua hasta el impacto. En cambio, la velocidad lateral (debida al efecto del viento) se mantiene relativamente constante con un valor promedio cercano a **18.24 m/s**, lo que refleja una componente lateral persistente durante todo el vuelo.

Finalmente, la Figura 5.11 muestra el **desplazamiento lateral total** del cohete en función del tiempo. Este valor alcanza un máximo de **1487.4 m**, lo que indica que el cohete recorrió una distancia lateral considerable desde el punto de lanzamiento debido a la influencia del viento. Esta métrica resulta clave para la predicción del lugar de impacto y la seguridad de la misión.

Estos resultados serán comparados posteriormente con los datos obtenidos del vuelo real, a fin de validar el modelo numérico empleado. En esta simulación los paracaídas son desactivados para permitir la comparación con los datos experimentales.

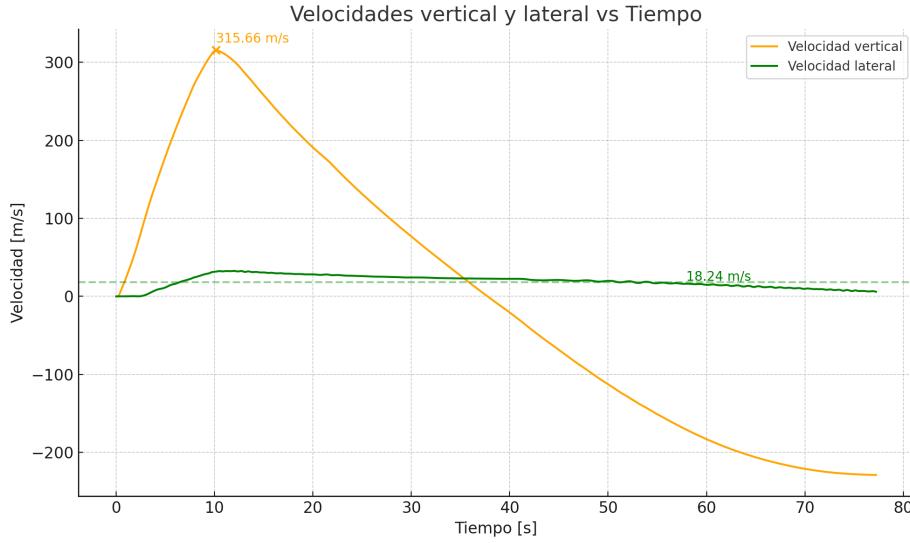


Figura 5.10: Velocidades vertical y lateral del cohete durante el vuelo simulado con OpenRocket

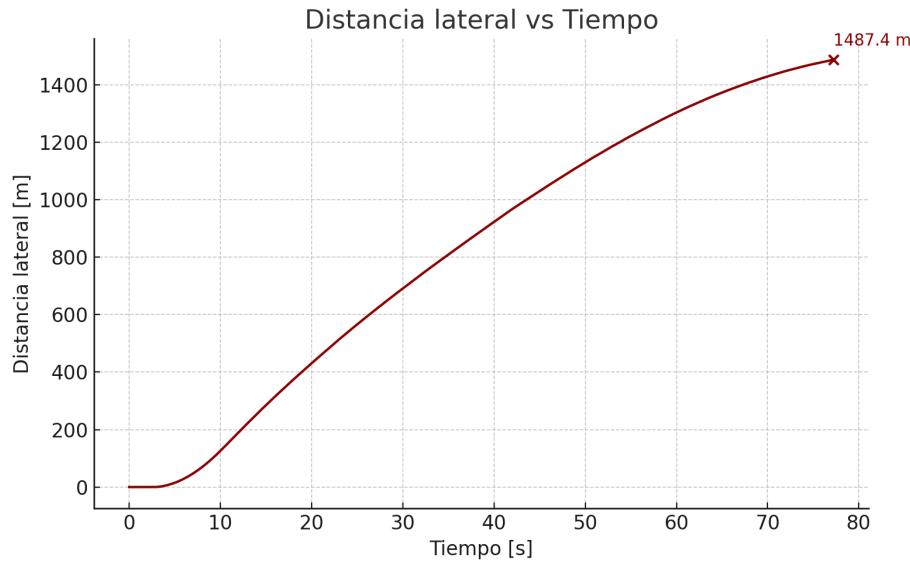


Figura 5.11: Desplazamiento lateral del cohete durante la simulación en OpenRocket

## 5.5. Vuelo simulado

El modelo y simulación desarrollados en este trabajo permiten analizar el comportamiento dinámico del cohete *Xitle II*. A continuación, se presentan los resultados más relevantes obtenidos con el integrador Dormand-Prince 8(5,3).

La evolución de la posición en las tres direcciones espaciales se muestra en la Figura 5.12. La componente vertical  $z(t)$  presenta una trayectoria parabólica característica, alcanzando una **altitud máxima** de **6577.64 m** a los **41.01 s**. La posición  $x(t)$

crece de forma casi lineal hasta un valor final de **8992.08 m**, reflejando el alcance horizontal inducido por el ángulo inicial de inclinación y el efecto del viento. Por otro lado, la componente  $y(t)$  apenas varía, con un máximo de **2.25 m**, lo que confirma que el vuelo se mantiene esencialmente dentro de un plano vertical.

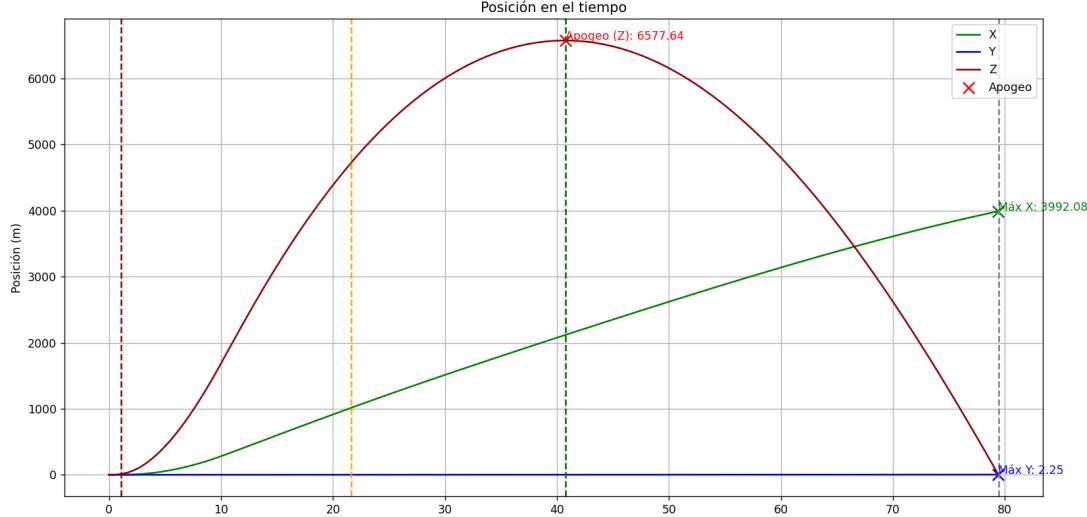


Figura 5.12: Evolución de la posición en las tres direcciones espaciales.

En cuanto a la velocidad, la Figura 5.13 muestra las componentes  $v_x(t)$ ,  $v_y(t)$  y  $v_z(t)$ . La componente vertical  $v_z$  domina el ascenso, alcanzando una **velocidad máxima** de **308.48 m/s** durante la fase propulsada, poco antes del MECO. Las componentes  $v_x$  y  $v_y$  son menores en magnitud, con máximos de **64.23 m/s** y **0.05 m/s** respectivamente.

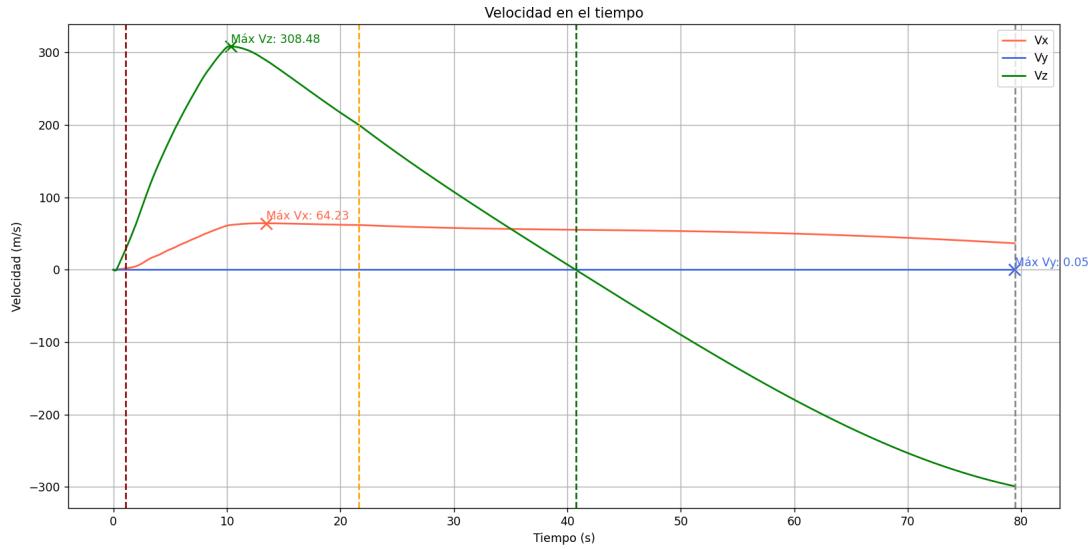


Figura 5.13: Evolución de las componentes de la velocidad durante el vuelo simulado.

La trayectoria del cohete se visualiza en la Figura 5.14, donde se presenta tanto una vista tridimensional como una proyección sobre el plano  $x-z$ . Se marcan los eventos

clave del vuelo: lanzamiento, MECO, apogeo e impacto. Se observa una trayectoria suave y coherente con el perfil de una fase impulsada seguida por caída balística.

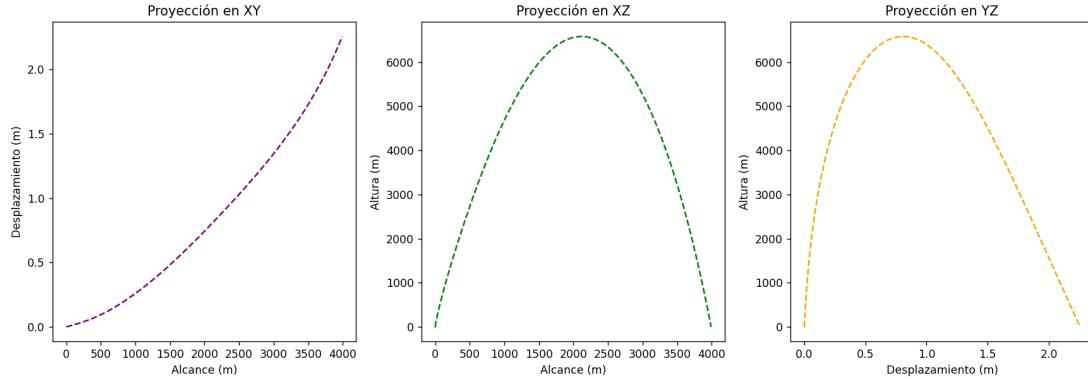


Figura 5.14: Trayectoria proyectada del cohete en los planos espaciales.

Respecto a la dinámica angular, el ángulo de inclinación  $\theta(t)$  disminuye progresivamente tras el MECO, como se muestra en la Figura 5.15. Las oscilaciones posteriores, producto de perturbaciones aerodinámicas y de la inercia del sistema, se ven amortiguadas por los momentos disipativos modelados.

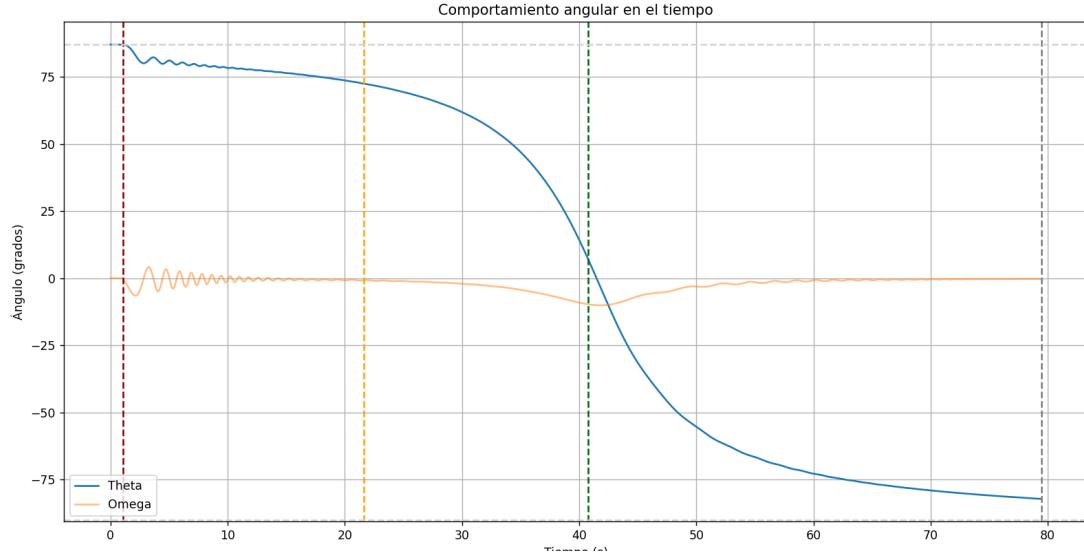


Figura 5.15: Evolución del ángulo de inclinación y velocidad angular.

Las gráficas de fuerzas aerodinámicas (Figura 5.16) confirman el predominio del empuje durante el ascenso inicial, así como la aparición de fuerzas de arrastre y normales durante la caída libre. El perfil temporal coincide con la variación de masa y con las fases de vuelo simuladas.

Por último, la Figura 5.17 muestra la dinámica angular completa. La aceleración angular presenta un pico poco después del MECO, con un valor máximo de aproximadamente  $115 \text{ rad/s}^2$ , seguido por oscilaciones que disminuyen en amplitud con el tiempo.

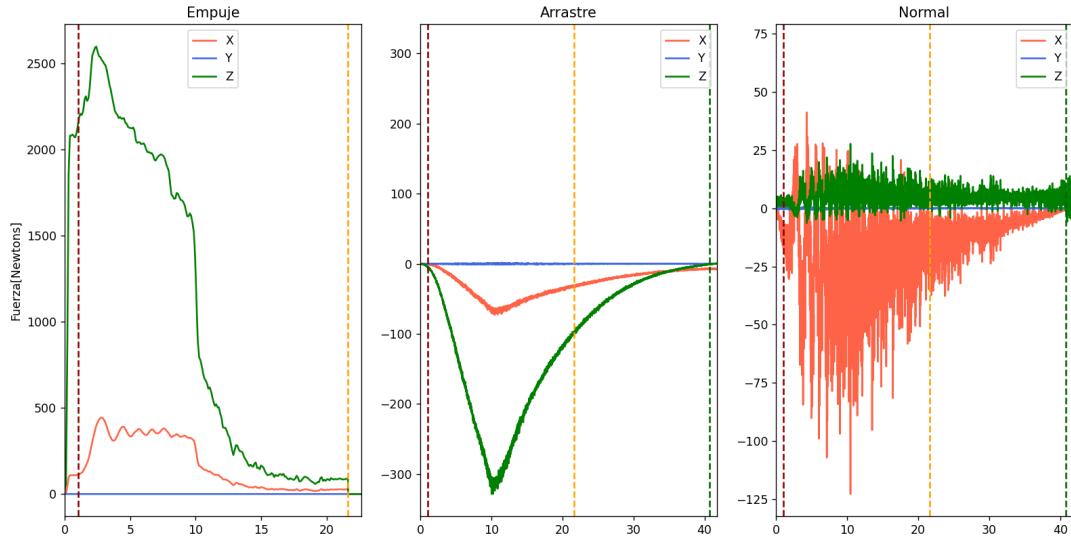


Figura 5.16: Componentes del empuje, arrastre y fuerza normal durante el vuelo.

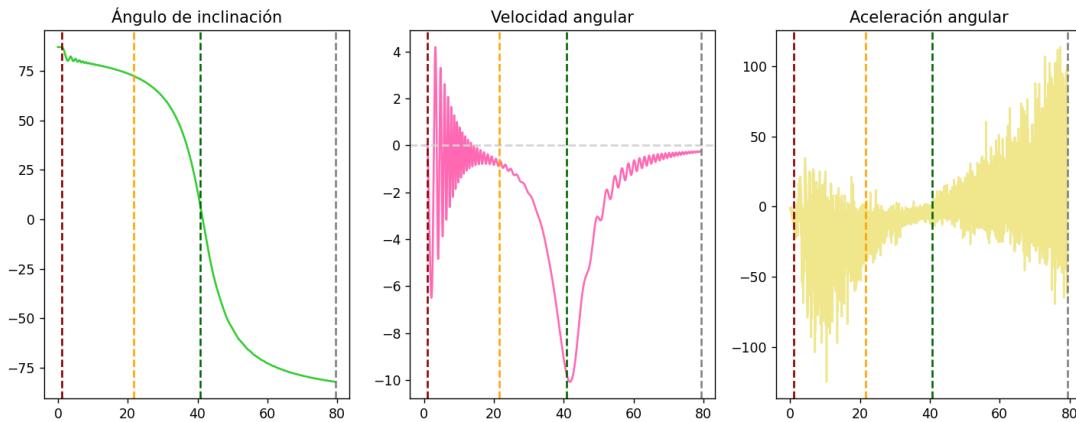


Figura 5.17: Evolución del ángulo, velocidad y aceleración angular.

La coherencia general del modelo se refleja también en los ángulos aerodinámicos. En la Figura 5.18, el ángulo de ataque se mantiene bajo durante el ascenso motorizado, y las variaciones posteriores validan la acción estabilizadora de los momentos modelados.

Adicionalmente, se generaron animaciones que ilustran visualmente el comportamiento dinámico del cohete a lo largo del vuelo. Estas visualizaciones fueron construidas a partir de los resultados de la simulación y permiten apreciar con mayor claridad la evolución temporal tanto de la orientación angular como de la trayectoria espacial.

En la primera animación ,Fig. 5.19, se visualiza la trayectoria 3D completa, evidenciando el ascenso vertical, el desplazamiento horizontal inducido y el retorno al plano del lanzamiento.

En la segunda, Fig. 5.20, se observa la rotación del vehículo en el espacio, destacando

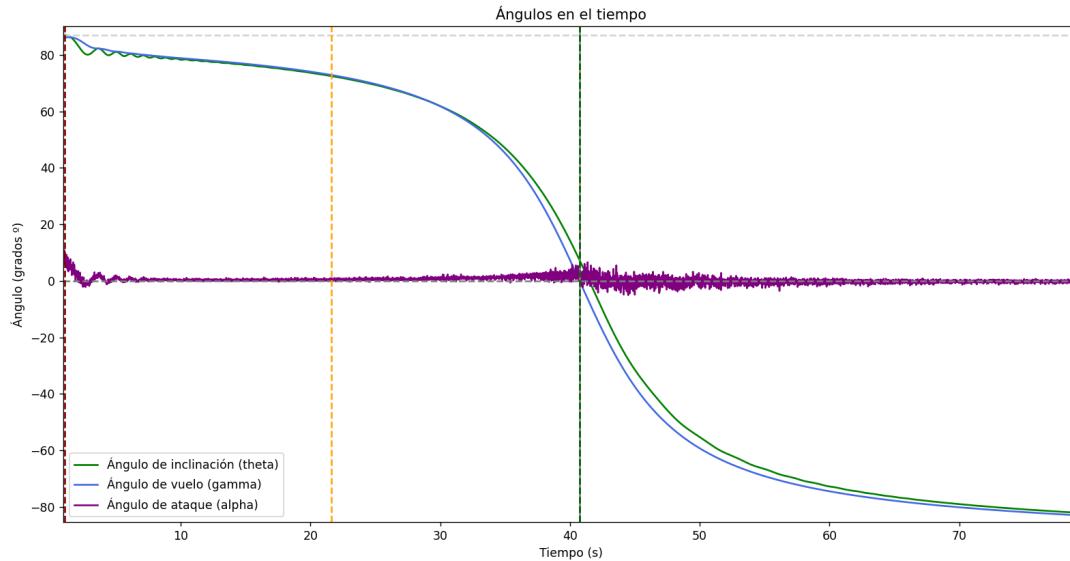


Figura 5.18: Ángulos característicos del vuelo simulado: inclinación ( $\theta$ ), trayectoria ( $\gamma$ ) y ataque ( $\alpha$ ).

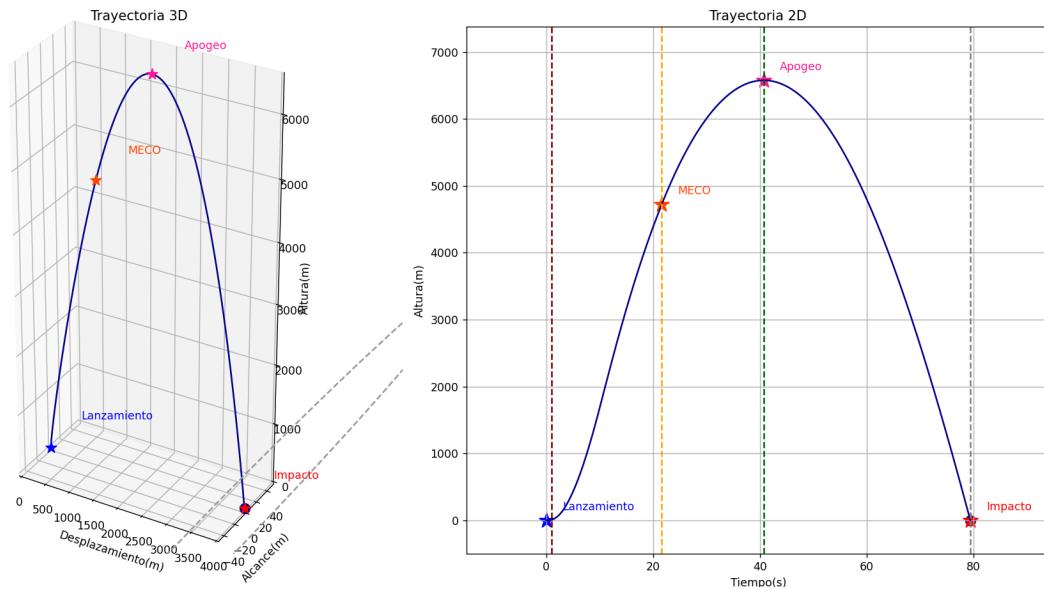


Figura 5.19: Trayectoria del vuelo de prueba

las oscilaciones alrededor del eje longitudinal tras el MECO.

Estas herramientas visuales complementan el análisis y son especialmente útiles para fines de validación.

En conjunto, los resultados obtenidos permiten caracterizar de forma integral el vuelo del cohete *Xtitle 2*. La evolución de variables como velocidad, posición, masa y ángulos, así como la interacción entre dinámica lineal y angular, confirma que el modelo reproduce fielmente las fases clave del vuelo: despegue, ascenso propulsado, apagado

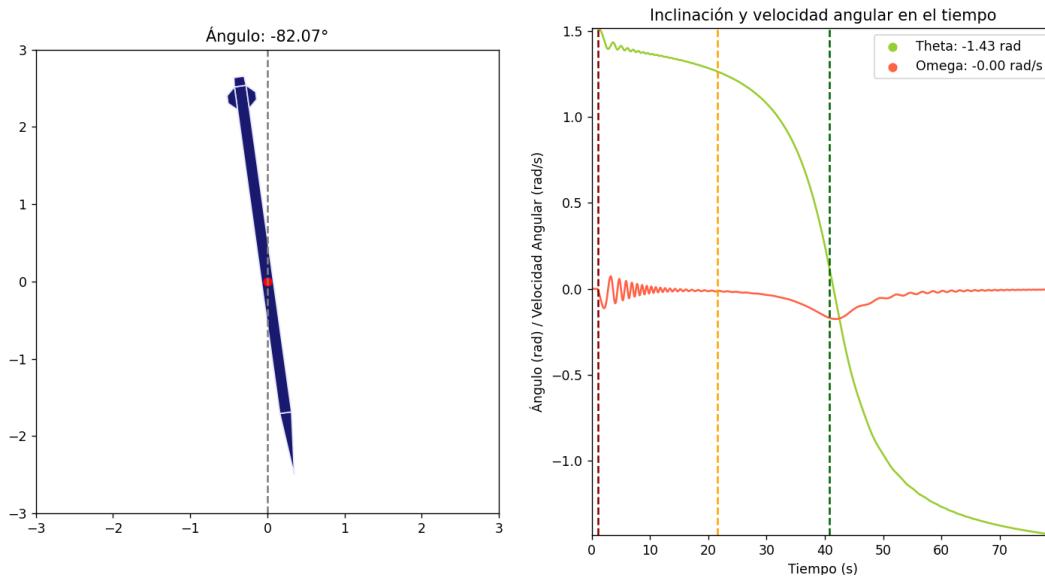


Figura 5.20: Movimiento angular en vuelo de prueba

del motor, apogeo, oscilaciones balísticas y descenso.

## 5.6. Verificación

La verificación consiste en comprobar que el modelo y el simulador están correctamente implementados, es decir, que resuelven de forma adecuada las ecuaciones planteadas. Resuelve la pregunta ¿Se está simulando correctamente?

### 5.6.1. Comparación de resultados

Este apartado tiene como objetivo comparar los resultados obtenidos a partir de los datos experimentales recolectados durante el vuelo real del cohete con aquellos generados mediante el simulador numérico desarrollado en este trabajo y el software OpenRocket. Esta comparación permite evaluar la fidelidad de ambos modelos en la predicción del comportamiento del vehículo, especialmente en términos de altitud alcanzada, desplazamientos, velocidades características y tiempos clave del perfil de vuelo.

La Figura 5.21 presenta la evolución de la altitud en función del tiempo para los tres conjuntos de datos: experimental (GPS FTW), simulación numérica de este trabajo y simulación con OpenRocket. Se observa que el simulador desarrollado en este trabajo logra una predicción sumamente precisa del perfil de ascenso, con una diferencia relativa de apenas 0.07 % en la altitud máxima y menor al 2.5 % en el tiempo de apogeo. En contraste, OpenRocket subestima significativamente la altitud alcanzada,

con un error relativo de  $-10.46\%$ , y adelanta la predicción del apogeo en más del  $8\%$ .

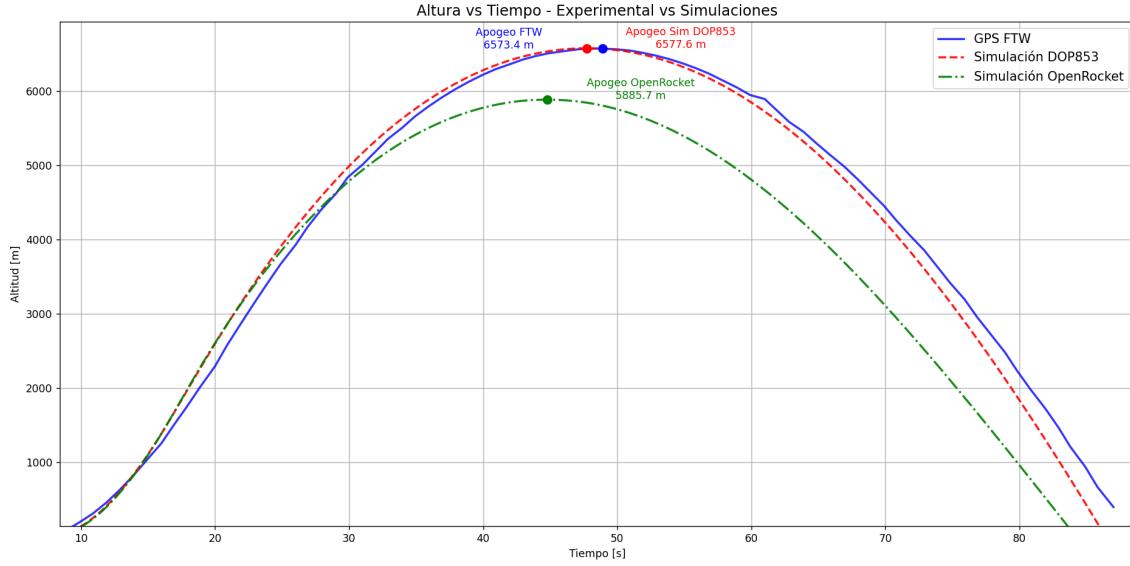


Figura 5.21: Evolución de la altura en vuelo experimental, simulado y de OpenRocket

La Figura 5.22 muestra la evolución de las velocidades vertical y horizontal. Se observa que tanto el simulador de este trabajo como OpenRocket sobreestiman la velocidad vertical máxima respecto a la medición experimental, con errores del orden del  $20\%$  y  $22.85\%$ , respectivamente. En cuanto a la velocidad horizontal, el simulador de la presente tesis proporciona una mejor aproximación con un error del  $-9.33\%$ , mientras que OpenRocket presenta una subestimación considerable, con un error mayor al  $-53\%$ .

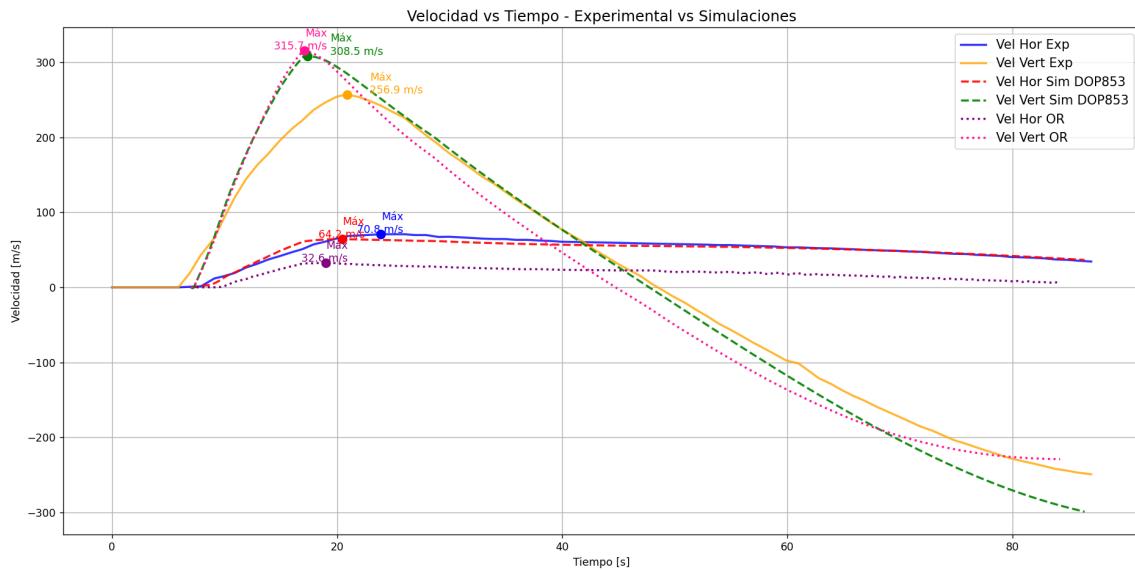


Figura 5.22: Evolución de la velocidad en vuelo experimental, simulado y de OpenRocket

Una diferencia particularmente notable aparece en el desplazamiento horizontal total, ilustrado en la Figura 5.23. El simulador aquí desarrollado estima un desplazamiento de 3992 metros, con un error relativo de apenas +6.06 % respecto al dato experimental. Por otro lado, OpenRocket subestima fuertemente esta magnitud, con una diferencia superior al 60 %, lo cual puede ser crítico para el análisis de zonas de impacto.

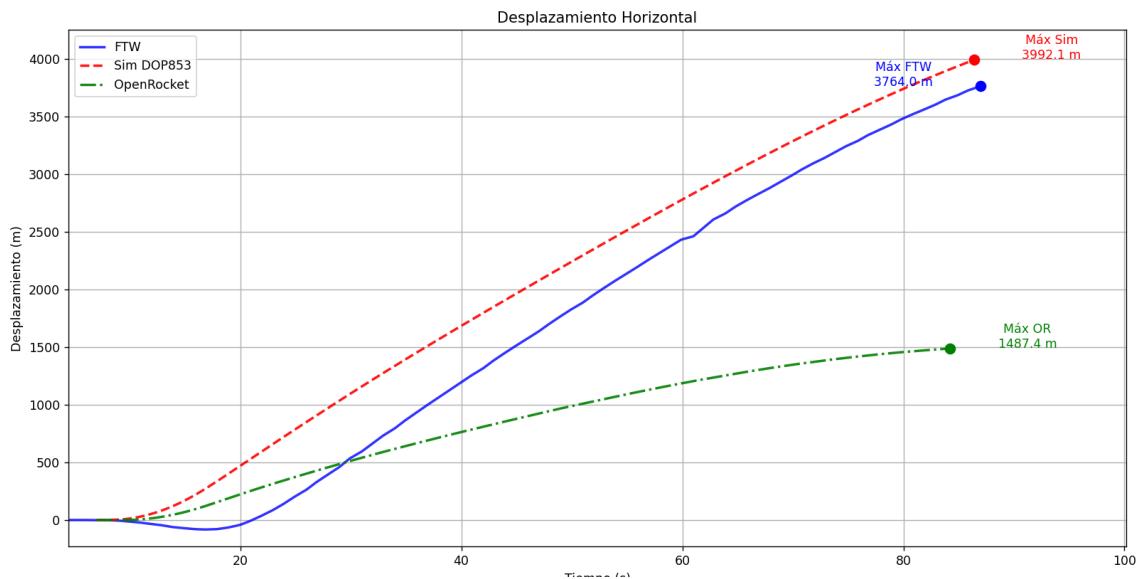


Figura 5.23: Evolución del desplazamiento horizontal en vuelo experimental, simulado y de OpenRocket

A continuación, se presenta la Tabla 5.4 que resume los valores más relevantes del vuelo, como la altitud máxima, tiempos de apogeo e impacto, velocidades máximas, desplazamiento y velocidad de impacto. Estos indicadores permiten visualizar el desempeño general del cohete y constituyen métricas clave para el diseño y evaluación de misiones reales.

Magnitud	Experimental	Sim propia	OpenRocket
Altitud máxima (m)	6573.36	6577.64	5885.68
Desplazamiento total (m)	3764.04	3992.08	1487.40
Tiempo de apogeo (s)	48.90	47.73	44.76
Tiempo de impacto (s)	87.00	86.41	84.23
Velocidad vertical máxima (m/s)	256.94	308.48	315.66
Velocidad horizontal máxima (m/s)	70.83	64.23	32.64
Velocidad vertical de impacto (m/s)	-249.17	-299.00	-229.02

Tabla 5.4: Resultados clave del vuelo experimental, simulado y de OpenRocket

Para cuantificar la precisión de las simulaciones, se calcularon los errores relativos respecto al experimento, como se muestra en la Tabla 5.5.

Magnitud	Error Sim propia (%)	Error OpenRocket (%)
Altitud máxima	+0.07	-10.46
Desplazamiento total	+6.06	-60.48
Tiempo de apogeo	-2.39	-8.47
Tiempo de impacto	-0.67	-3.19
Velocidad vertical máxima	+20.06	+22.85
Velocidad horizontal máxima	-9.33	-53.92

Tabla 5.5: Errores relativos respecto al vuelo experimental

Estos resultados confirman que el simulador desarrollado en esta tesis ofrece una aproximación mucho más precisa al vuelo real en comparación con herramientas comerciales como OpenRocket. La incorporación de modelos aerodinámicos detallados, condiciones iniciales personalizables y métodos numéricos adaptativos permite simular con realismo el comportamiento del cohete durante su vuelo, desde el despegue hasta el impacto. Esta capacidad predictiva representa una herramienta útil tanto para el diseño iterativo de vehículos como para la planificación segura de futuras misiones.

## 5.7. Validación

¿Estamos simulando el vuelo adecuadamente? La validación busca demostrar que el modelo representa fielmente el comportamiento del sistema real. Para ello, se usaron diferentes configuraciones de vuelo y se compararon con los datos experimentales.

En esta sección se realizan simulaciones de prueba de situaciones significativas con el vehículo *Xitle II* para corroborar que los resultados tienen un significado lógico coherente y así aumentar nuestra confianza en el programa.

Los valores de referencia empleados en las simulaciones corresponden a los datos resumidos anteriormente. Estas pruebas se hacen desde el punto de vista del usuario en un contexto lo más cercano posible al real.

### 5.7.1. Prueba 1. Vuelo CON paracaídas

Con el objetivo de verificar la correcta simulación de un sistema de recuperación pasiva, se implementó el despliegue de un paracaídas tipo *drogue* durante la fase de descenso. Esta configuración permite evaluar el comportamiento dinámico del vehículo una vez alcanzado el apogeo, así como su desaceleración progresiva hasta el impacto.

En la Figura 5.24 se presentan las trayectorias 3D y 2D del vuelo simulado con paracaídas. Se distinguen claramente los eventos clave: lanzamiento, apagado del motor (*MECO*), el apogeo, la activación del sistema de recuperación y el impacto controlado.

La fase ascendente se desarrolla de forma nominal hasta alcanzar la altitud máxima. Una vez superado el apogeo, se activa el sistema de recuperación, dando lugar a una fase de caída alargada en el tiempo.

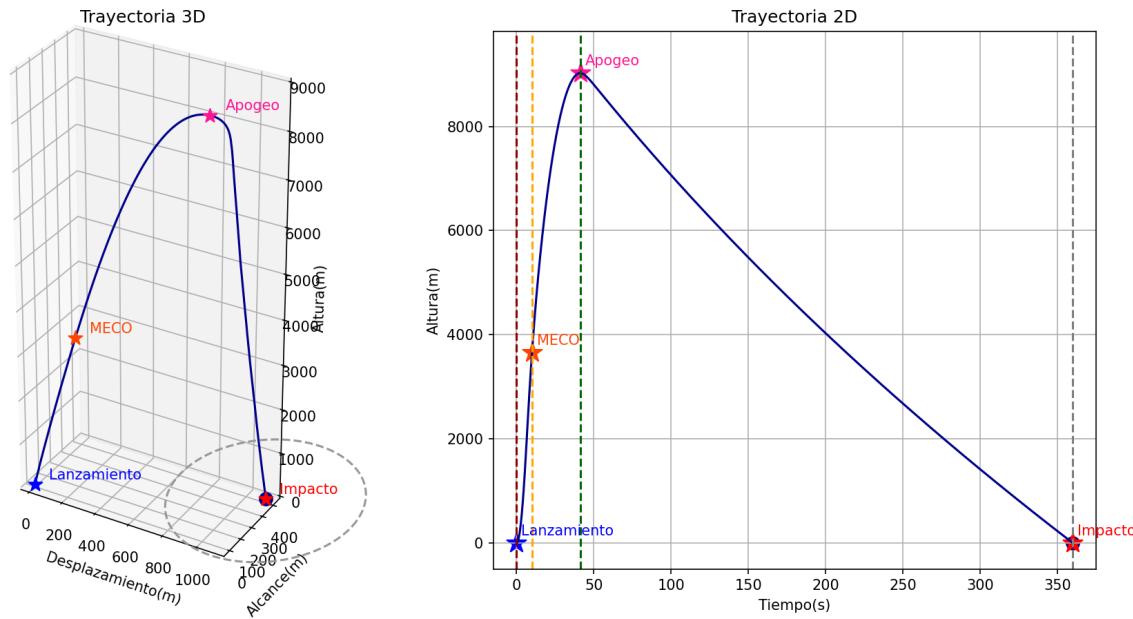


Figura 5.24: Trayectoria con paracaídas

La trayectoria descendente muestra una marcada desaceleración, evidenciada por la curvatura extendida de la parábola en la gráfica temporal. El impacto se produce a velocidades reducidas y con una pendiente suavizada, confirmando la efectividad del paracaídas para disminuir la energía cinética del vehículo al momento del aterrizaje.

Este experimento demuestra que el simulador es capaz de modelar con precisión tanto la activación como los efectos dinámicos de un sistema de recuperación, y representa una base sólida para futuras extensiones que incluyan múltiples etapas de apertura o sistemas duales.

### 5.7.2. Prueba 2. Vehículo inestable

Con el fin de demostrar la capacidad del simulador para detectar configuraciones inestables, se diseñó una prueba alterando exclusivamente la distribución de masas internas del cohete *Xtitle II*, sin modificar su geometría externa. Específicamente, se desplazaron masas hacia la parte trasera del vehículo, generando una inversión en la relación entre el centro de gravedad (CG) y el centro de presión (CP). Esta modificación produjo un vehículo aerodinámicamente inestable.

La Figura 5.25 muestra un fotograma del comportamiento del cohete simulado bajo esta condición. Se observa que, tras el lanzamiento, el ángulo de inclinación comienza a incrementarse de manera sostenida y no se amortigua, generando una trayectoria

oscilante e incontrolada. La velocidad angular también crece conforme se aleja del equilibrio, lo que evidencia la inestabilidad dinámica provocada.

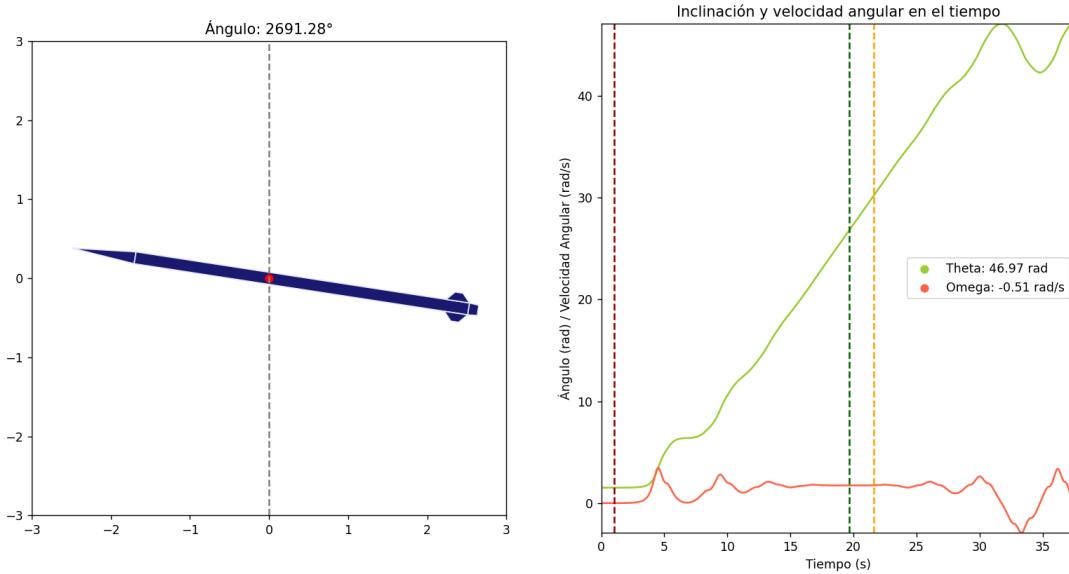


Figura 5.25: Resultado de la simulación de un cohete inestable. El ángulo se incrementa sin amortiguamiento debido a la inversión CG–CP.

La Tabla 5.6 presenta los valores precisos del CG y del CP para ambos modelos. En el caso original, el CP se encuentra por detrás del CG, generando un momento restaurador ante perturbaciones. En contraste, la configuración modificada presenta el CP adelantado, lo cual provoca momentos desestabilizantes.

Vehículo	CG [m]	CP [m]	CP - CG [m]
Original (Xitle)	2.524	3.134	+0.611
Modificado (inestable)	3.045	2.831	-0.214

Tabla 5.6: Comparación del centro de gravedad (CG) y centro de presión (CP) entre el modelo original y el modificado.

Esta prueba confirma que el modelo no solo es capaz de simular trayectorias nominales, sino también de identificar con precisión configuraciones que conducen a inestabilidad. De este modo, se valida su utilidad como herramienta predictiva para el diseño seguro de vehículos aeroespaciales.

### 5.7.3. Prueba 3. Simulación con y sin viento

En esta prueba se analizó el impacto de la presencia de viento en la trayectoria y velocidad del cohete. Para ello se realizaron dos simulaciones con las mismas condiciones iniciales y parámetros físicos del vehículo, variando únicamente la activación

del modelo de viento. Se utilizó el generador Viento3D, configurado con valores representativos de un entorno de viento moderado a fuerte, considerando las condiciones atmosféricas características del sitio de lanzamiento en la **Spaceport America Cup**, ubicado en el desierto de Nuevo México.

Los parámetros elegidos para esta simulación fueron:

- Viento base: 5 m/s
- Ráfagas con media: 3 m/s
- Desviación de ráfagas: 2 m/s
- Dirección media: 230°
- Variación angular:  $\pm 10^\circ$

Estas condiciones fueron seleccionadas con base en reportes meteorológicos históricos y observaciones de campo, donde se ha documentado la presencia frecuente de ráfagas impredecibles, alta exposición solar y flujo de viento persistente debido a la topografía abierta del terreno. El modelo de viento se actualiza aleatoriamente en cada paso del integrador y genera un vector tridimensional que modifica directamente la velocidad relativa del aire y, por lo tanto, las fuerzas aerodinámicas aplicadas al vehículo.

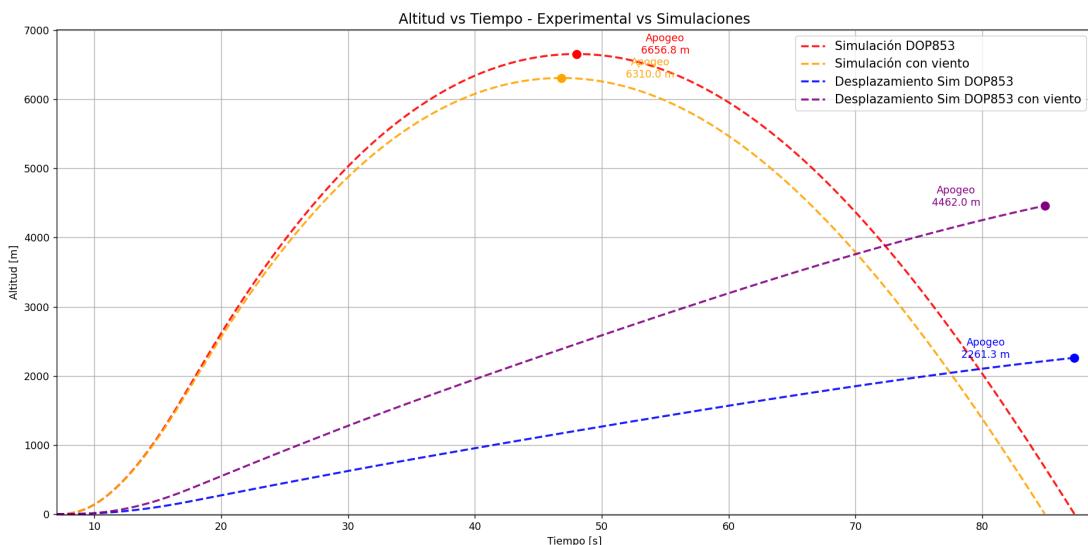


Figura 5.26: Comparación de la altitud y desplazamiento horizontal con y sin viento.

La Figura 5.26 muestra el efecto del viento sobre la trayectoria del cohete. Se observa que, aunque la altitud máxima disminuye en más de 300 metros debido a la orientación desfavorable del viento, el impacto más significativo se produce en el desplazamiento horizontal. La simulación con viento alcanza un desplazamiento más del doble que la simulación sin viento, lo que pone de manifiesto la importancia de considerar la acción del viento en el análisis de seguridad de la zona de caída.

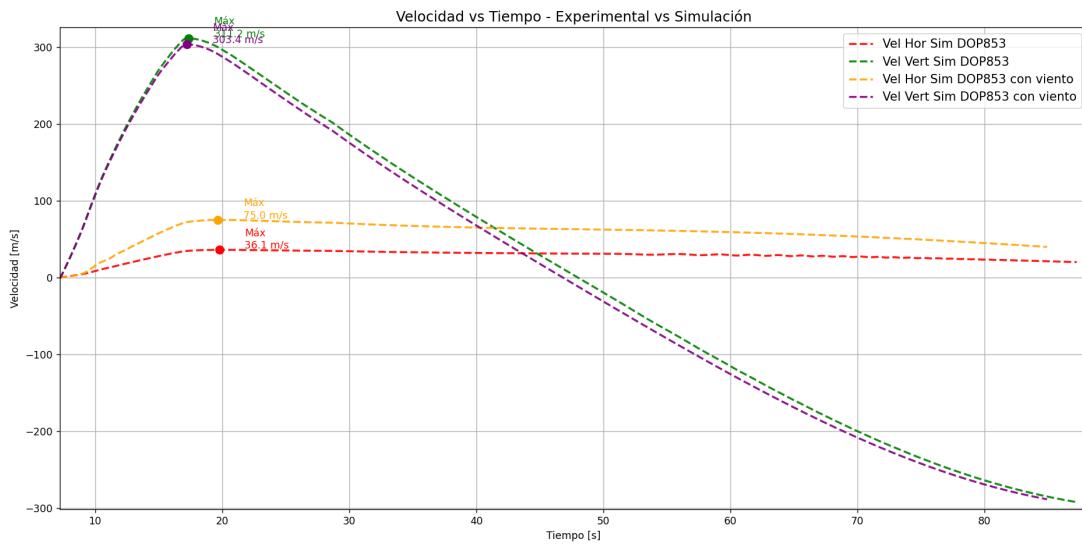


Figura 5.27: Comparación de velocidades verticales y horizontales con y sin viento.

En cuanto a las velocidades, la Figura 5.27 muestra que el viento induce una aceleración lateral significativa en los primeros segundos del vuelo, duplicando la velocidad horizontal máxima. La velocidad vertical también se ve ligeramente afectada, reduciendo su valor máximo y modificando la dinámica de ascenso.

Para analizar en mayor detalle el comportamiento del viento introducido, se generaron las siguientes gráficas:

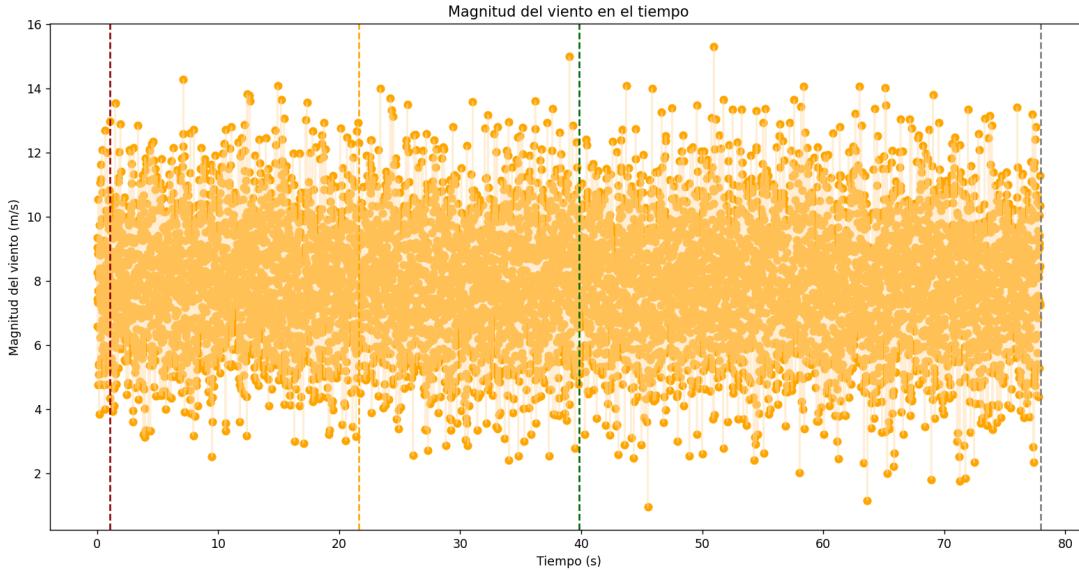


Figura 5.28: Magnitud del viento a lo largo del tiempo durante la simulación.

En la Figura 5.28, se observa que la magnitud total del viento varía entre 4 y 15 m/s, producto de la suma de la componente base con ráfagas gaussianas aleatorias. Por su parte, la Figura 5.29 muestra que la dirección del viento se mantiene centrada

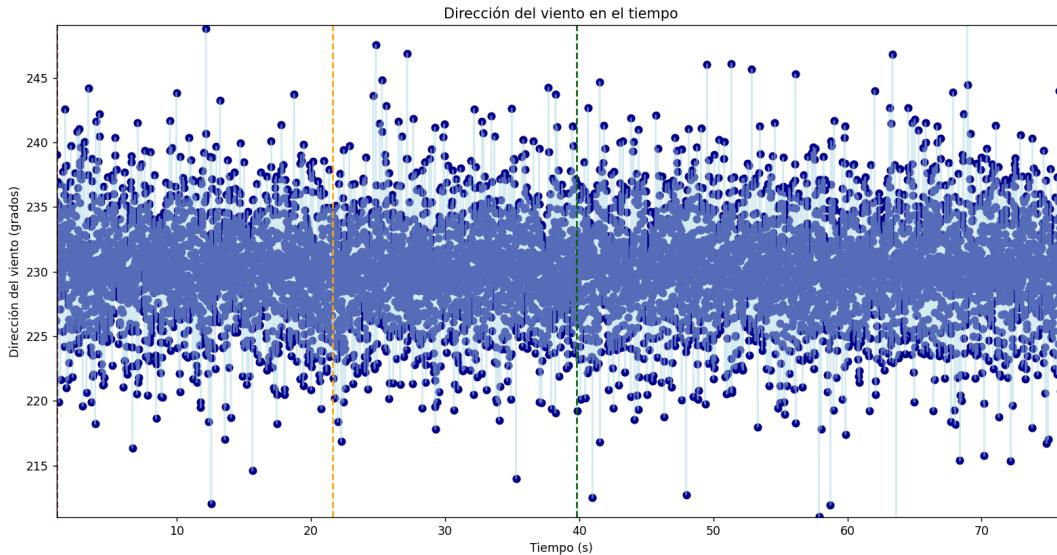


Figura 5.29: Variación de la dirección del viento en el tiempo.

alrededor de  $230^\circ$ , con una dispersión típica de  $\pm 10^\circ$  que genera una leve oscilación en el vector de fuerza aerodinámica lateral.

Tabla 5.7: Comparación de resultados clave con y sin viento

Magnitud	Sin viento	Con viento
Altitud máxima (m)	6656.8	6310.0
Velocidad vertical máxima (m/s)	311.2	303.4
Velocidad horizontal máxima (m/s)	36.1	75.0
Tiempo de apogeo (s)	48.0	47.1
Tiempo de impacto (s)	87.3	86.2
Desplazamiento total (m)	2261.3	4462.0

La Tabla 5.7 resume cuantitativamente las diferencias observadas. Se evidencia que la mayor alteración se produce en el desplazamiento horizontal (incremento del 97.3 %), seguido por la reducción de la altitud máxima (-5.2 %). Aunque los tiempos de vuelo se ven ligeramente afectados, los valores finales permiten concluir que el viento representa una fuente importante de error en la estimación de zonas de impacto, por lo que su inclusión en simulaciones de diseño y seguridad es fundamental.

#### 5.7.4. Prueba 4. Variación de parámetros de diseño

Se variaron las curvas de empuje, de coeficiente de arrastre y de variación de masa. También, se modificó la distribución interna de masa del cohete, observando que el simulador logró predecir condiciones distintas a las nominales, demostrando su capacidad predictiva.

La variación de parámetros de diseño nos permite analizar cómo cambios en variables clave, como el centro de gravedad, el área de las aletas o la longitud del fuselaje, afectan la estabilidad y el desempeño del vehículo. Este enfoque facilita la optimización del diseño para cumplir con los objetivos específicos de la misión.

# 6 Conclusiones

A lo largo de este trabajo no solo se modeló el fenómeno físico de interés, sino que se programó un **software para la resolución numérica de problemas de valor inicial (PVI)** de la forma

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t); \boldsymbol{\theta}), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \quad (6.1)$$

donde la función derivada  $\mathbf{f}$  representa la dinámica del sistema bajo estudio y  $\boldsymbol{\theta}$  agrupa los parámetros físicos o de modelo. Aunque la motivación central fue **reproducir trayectorias de vuelo de cohetes suborbitales**, el *diseño orientado a objetos* —con la clase `Vuelo` encapsulando  $\mathbf{f}$ — permite sustituir esa clase por cualquier otra que defina un nuevo campo vectorial. Así, el **software numérico** puede:

- predecir la evolución de carteras financieras (p. ej. el sistema de Black–Scholes en su versión determinista),
- modelar brotes epidemiológicos (SIR, SEIR con vacunación),
- estudiar poblaciones biológicas, circuitos eléctricos y modelos climáticos de baja dimensión,

con solo modificar una decena de líneas de código y sin alterar el núcleo numérico.

Teniendo como resultado un simulador numérico para trayectorias de vuelo de cohetes suborbitales, que estima el comportamiento dinámico de vehículos bajo distintas condiciones iniciales y ambientales. El simulador desarrollado permitió integrar múltiples aspectos del vuelo real, tales como la variación de empuje, la masa dependiente del tiempo, los efectos del viento y los momentos aerodinámicos, estabilizadores y amortiguadores.

Asimismo, se incorporó un conjunto de integradores numéricos que fueron comparados en términos de precisión, estabilidad y eficiencia, destacando finalmente el método Dormand-Prince (DOP853) como el más adecuado para las necesidades del sistema.

Uno de los principales hallazgos es que la simulación numérica, al permitir la variación sistemática de parámetros de diseño y condiciones de vuelo, se convierte en una herramienta poderosa para la toma de decisiones en la etapa temprana del desarrollo de un vehículo aeroespacial. Aspectos como el empuje del motor, la distribución de masas, el ángulo de lanzamiento o la presencia de viento, pueden analizarse en detalle

para predecir su impacto en la trayectoria, mejorar el diseño del sistema, y anticipar zonas seguras de caída.

Como ya mencionamos, la simulación, además de ser aplicable en múltiples áreas del conocimiento, tiene la ventaja de reducir significativamente el costo, el riesgo y el tiempo asociado a la experimentación. Sin embargo, es fundamental comprender las limitaciones inherentes al modelo utilizado. Toda simulación depende de las suposiciones que se hacen al modelar un fenómeno, por lo que es esencial asegurarse de que estas sean coherentes con el nivel de detalle requerido. Ignorar las limitaciones del modelo puede conducir a interpretaciones erróneas o a decisiones comprometidas.

En particular, este trabajo pone en evidencia la diferencia entre las escalas de tiempo involucradas en la dinámica traslacional y rotacional del vehículo. Mientras que el momento lineal cambia lentamente, en escalas del orden de varios segundos, el momento rotacional puede experimentar variaciones importantes en fracciones de segundo. Esta diferencia plantea un reto para la simulación numérica, pues exige un integrador lo suficientemente preciso como para capturar ambos comportamientos sin comprometer la estabilidad del sistema.

La integración adecuada de la simulación dentro del proceso de análisis y síntesis proporciona una plataforma virtual segura y controlada para explorar distintos escenarios, evaluar estrategias de diseño y mejorar la planificación de misiones. Esta capacidad resulta especialmente valiosa en contextos donde las pruebas experimentales son costosas o implican riesgos considerables, como es el caso de la cohetería experimental.

No obstante, la simulación no sustituye la experimentación ni garantiza por sí sola la veracidad de los resultados. La elección del modelo matemático, la validación frente a datos reales, el análisis de sensibilidad, la documentación transparente del proceso y la interpretación crítica de los resultados son pasos indispensables para garantizar la confiabilidad de las conclusiones obtenidas.

# Bibliografía

- [isa, 1971] (1971). U.s. standard atmosphere, 1971. Technical report, U.S. Government Printing Office, Washington, D.C. NOAA, NASA, USAF.
- [AEM, 2010] (2010). Ley que crea la agencia espacial mexicana. Diario Oficial de la Federación.
- [Alonso y Coria, 1977] Alonso y Coria, A. (1977). *Apuntes DCR-1: Dinámica de Cohetes y Rentrée Atmosférica*. Facultad de Ciencias, UNAM, Ciudad de México.
- [Alonso y Coria and García, 1975] Alonso y Coria, A. and García, P. (1975). Diseño de un espectrómetro mössbauer para aleaciones de alta temperatura. *Revista Mexicana de Física*, 21(3):145–152.
- [Arauz Lara, 2022] Arauz Lara, B. J. L. (2022). Cabo tuna: A 65 años del nacimiento de la era espacial mexicana. Reportaje en línea.
- [Aspirespace, 2011a] Aspirespace (2011a). A dynamic stability analysis rocket simulator. <http://www.aspirespace.org.uk/downloads/A%20dynamic%20stability%20rocket%20simulator.pdf>. [Accessed 03-04-2024].
- [Aspirespace, 2011b] Aspirespace (2011b). Rocket aerodynamics. <http://www.aspirespace.org.uk/downloads/A%20dynamic%20stability%20rocket%20simulator.pdf>. [Accessed 03-04-2024].
- [Barrowman, 1966] Barrowman, J. (1966). The theoretical prediction of the center of pressure.
- [Burden and Faires, 2021] Burden, R. L. and Faires, J. D. (2021). *Numerical Analysis*. Cengage Learning, Boston, 11th edition.
- [Butcher, 2016] Butcher, J. C. (2016). *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Chichester, 3rd edition.
- [Carbajal, 2024] Carbajal, O. A. C. (2024). Desarrollo del sistema de alimentación para un sistema de propulsión líquida de un cohete suborbital.
- [Ceotto Giovani H., 2021] Ceotto Giovani H., Schmitt Rodrigo N., A. G. F. P. L. A. C. B. S. (2021). Rocketpy: Six degree-of-freedom rocket trajectory simulator. *Journal of Aerospace Engineering*.

- [Charles E. Rogers, 2019] Charles E. Rogers, D. C. (2019). Rogers aeroscience rasaero ii aerodynamic analysis and flight simulation program-users manual. Technical report, Rogers Aerospace.
- [de Comunicaciones y Transportes, 1964] de Comunicaciones y Transportes, S. (1964). Informe técnico anual 1964 de la comisión nacional del espacio exterior. Archivo Histórico de la SCT, México. Accedido 10 may 2025.
- [de Comunicaciones y Transportes, 2010] de Comunicaciones y Transportes, S. (2010). Norma oficial mexicana nom-009-sct2/2009. <https://www.dof.gob.mx/normasOficiales/3964/sct/sct.htm>. [Accessed 26-04-2024].
- [de ENMICE, 2023] de ENMICE, A. E. M.-C. (2023). Guía de procedimientos y operaciones estandarizadas. [Accessed 26-04-2024].
- [de Relaciones Exteriores, 2021] de Relaciones Exteriores, S. (2021). México se adhiere a los acuerdos artemisa. Comunicado de prensa.
- [Dormand and Prince, 1980] Dormand, J. R. and Prince, P. J. (1980). A family of embedded runge–kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26.
- [Duarte et al., 2022] Duarte, C., Chávez, R., and et al. (2022). Colmena: Mexican micro-robots for lunar exploration. In *IEEE Aerospace Conference*.
- [Gear, 1971] Gear, C. W. (1971). *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs.
- [Hairer and Wanner, 1996] Hairer, E. and Wanner, G. (1996). *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, volume 14 of *Springer Series in Computational Mathematics*. Springer, Berlin, 2nd edition.
- [Hindmarsh, 1983] Hindmarsh, A. C. (1983). Odepack, a systematized collection of ODE solvers. In Stepleman, R. S., editor, *Scientific Computing*, pages 55–64. North-Holland, Amsterdam. Describes LSODE, LSODA, LSODAR.
- [Kleppner and Kolenkow, 2014] Kleppner, D. and Kolenkow, R. (2014). *An Introduction to Mechanics*. Cambridge University Press.
- [Mexicana, 2025] Mexicana, A. E. (2025). Plan estratégico 2025: Consolidación de clústeres aeroespaciales. Documento interno.
- [Muñoz et al., 1973] Muñoz, C. R. D., Verdugo, M. A. F., García, H. C. H., Gómez, R. V. M. L., Cifuentes, A. S., and de J. Vega Jiménez, J. (1973). Diseño y construcción de un cohete de propelente líquido para sondeo atmosférico. Tesis de licenciatura, Facultad de Ingeniería, Universidad Nacional Autónoma de México, Ciudad Universitaria. Director del seminario: Ing. Dagoberto de la Serna Valdivia.
- [Niskanen, 2009] Niskanen, S. (2009). Development of an open source model rocket simulation software. Master's thesis, Helsinki University of Technology.

- [Niskanen, 2013] Niskanen, S. (2013). Openrocket technical documentation. Technical report.
- [Press et al., 2007] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 3rd edition.
- [Pérez, 2024] Pérez, E. S. P. (2024). Diseño, implementación y evaluación de sistema para monitoreo/encendido de motores cohete híbridos.
- [Roache, 1998] Roache, P. J. (1998). *Verification and Validation in Computational Sciences and Engineering*. Hermosa Publishers.
- [Rubio, 2023] Rubio, I. A. (2023). Diseño mecánico estructural del sistema de frenado aerodinámico implementado en el cohete xitle.
- [Shannon, 1988] Shannon, R. E. (1988). *Simulación de Sistemas*. Editorial Trillas México.
- [Simon Box, 2009] Simon Box, Christopher M. Bishop, H. H. (2009). *Estimating the dynamic and aerodynamic parameters of passively controlled high power rockets for flight simulaton*.
- [Simon Box, 2010] Simon Box, Christopher M. Bishop, H. H. (2010). *Stochastic Six-Degree-of-Freedom Flight Simulator for Passively Controlled High-Power Rockets*.
- [Smith et al., 2021] Smith, M., Hernández, J., and et al. (2021). Aztechsat-1 mission overview. *Acta Astronautica*, 181:50–57.
- [Sociedad Universitaria de Cohetes Experimentales, 1973] Sociedad Universitaria de Cohetes Experimentales (1973). *Manual de Propulsión de Pólvoras Compuestas*. SUCE, Ciudad Universitaria, UNAM. 125 pp. Tesis colectiva de miembros de la SUCE.
- [Teschl, 2012] Teschl, G. (2012). *Ordinary Differential Equations and Dynamical Systems*, volume 140 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence.
- [Tierno et al., 2012] Tierno, M., Cortés, M., and Márquez, C. (2012). *Mecánica del vuelo*. Ibergaceta.
- [Tim, 1997] Tim, V. M. (1997). Rocksim model rocket design and simulation software.
- [UNAM, 2024] UNAM, P. (2024). Xitle ii hybrid rocket project technical report.
- [UPAEP, 2024] UPAEP (2024). Presentación del cubesat gxiba-1. Rueda de prensa.
- [Ángeles Ramírez, 1908] Ángeles Ramírez, F. d. J. (1908). *Teoría del Tiro*. Talleres del Departamento de Estado Mayor, Ciudad de México. 1<sup>a</sup> ed., VIII + 138 pp.



# Índice de figuras

1.1.	Ciclo de diseño asistido por simulación en la Ingeniería Realizado con IA en Eraser.io . . . . .	2
1.2.	Proceso de Modelación . . . . .	4
1.3.	Estructura general de un cohete híbrido . . . . .	7
1.4.	Diagrama de fases de vuelo. Imagen modificada del reporte técnico de Propulsión UNAM . . . . .	8
1.5.	Equipo Propulsión UNAM y sus cohetes. Temporada 2023 . . . . .	15
1.6.	Momentos principales: Cabeceo, guiñada y alabeo . . . . .	16
1.7.	Principales softwares comerciales . . . . .	18
1.8.	Otros softwares comerciales . . . . .	20
2.1.	Diagrama de cuerpo libre y ángulos ( $\alpha$ : ángulo de ataque, $\gamma$ : ángulo de vuelo, $\theta$ : ángulo de inclinación o “pitch”) . . . . .	36
2.2.	Curva de Empuje del motor "Mega Punisher" . . . . .	40
2.3.	Vectores de velocidad del cohete, velocidad del viento y velocidad relativa . . . . .	41
2.4.	Variación del coeficiente de arrastre de acuerdo a la velocidad . . . . .	42
2.5.	Momento estabilizador en configuración no estable y estable . . . . .	46
3.1.	Interpolación de curva de empuje . . . . .	53
3.2.	Convergencia de los métodos en el Caso 1 . . . . .	68
3.3.	Eficiencia numérica de los métodos en el Caso 1 . . . . .	69
3.4.	Eficiencia computacional de los métodos en el Caso 1 . . . . .	70
3.5.	Convergencia de los métodos en el Caso 2 . . . . .	71
3.6.	Eficiencia numérica en función del número de pasos para el Caso 2. . . . .	72
3.7.	Evolución temporal del paso de tiempo en el Caso 2 . . . . .	73
3.8.	Convergencia de los métodos en el Caso 3 . . . . .	74
3.9.	Eficiencia numérica en el Caso 3 . . . . .	75
4.1.	Esquema general de la simulación . . . . .	81
4.2.	Diagrama de la función fun-derivs . . . . .	84
4.3.	Diagrama de flujo de simular vuelo . . . . .	85
4.4.	Módulo de definición del cohete en la interfaz gráfica. . . . .	89
4.5.	Importación y visualización de curvas de empuje y masa. . . . .	90
4.6.	Parámetros de simulación: entorno, viento, riel y método de integración. . . . .	90

5.1. Fotografías del cohete Xitle 2 en Spaceport America Cup 2024 . . . . .	96
5.2. CAD seccionado del cohete Xitle . . . . .	96
5.3. Curva de empuje- Prueba estática 28 de mayo, 2024 . . . . .	98
5.4. Interconexion General de los sistemas electronicos . . . . .	99
5.5. Visualización del Xitle 2 . . . . .	101
5.6. Altitud registrada por el GPS Featherweight. . . . .	102
5.7. Velocidades vertical y horizontal registradas por el GPS Featherweight.	102
5.8. Esquema del cohete Xitle en el programa Open Rocket . . . . .	103
5.9. Altura del cohete en función del tiempo durante la simulación en Open-Rocket . . . . .	103
5.10. Velocidades vertical y lateral del cohete durante el vuelo simulado con OpenRocket . . . . .	104
5.11. Desplazamiento lateral del cohete durante la simulación en OpenRocket	104
5.12. Evolución de la posición en las tres direcciones espaciales. . . . .	105
5.13. Evolución de las componentes de la velocidad durante el vuelo simulado.	105
5.14. Trayectoria proyectada del cohete en los planos espaciales. . . . .	106
5.15. Evolución del ángulo de inclinación y velocidad angular. . . . .	106
5.16. Componentes del empuje, arrastre y fuerza normal durante el vuelo. .	107
5.17. Evolución del ángulo, velocidad y aceleración angular. . . . .	107
5.18. Ángulos característicos del vuelo simulado: inclinación ( $\theta$ ), trayectoria ( $\gamma$ ) y ataque ( $\alpha$ ). . . . .	108
5.19. Trayectoria del vuelo de prueba . . . . .	108
5.20. Movimiento angular en vuelo de prueba . . . . .	109
5.21. Evolución de la altura en vuelo experimental, simulado y de OpenRocket	110
5.22. Evolución de la velocidad en vuelo experimental, simulado y de Open-Rocket . . . . .	110
5.23. Evolución del desplazamiento horizontal en vuelo experimental, simulado y de OpenRocket . . . . .	111
5.24. Trayectoria con paracaídas . . . . .	113
5.25. Resultado de la simulación de un cohete inestable. El ángulo se incrementa sin amortiguamiento debido a la inversión CG–CP. . . . .	114
5.26. Comparación de la altitud y desplazamiento horizontal con y sin viento.	115
5.27. Comparación de velocidades verticales y horizontales con y sin viento.	116
5.28. Magnitud del viento a lo largo del tiempo durante la simulación. . . .	116
5.29. Variación de la dirección del viento en el tiempo. . . . .	117
A.1. Representación general del movimiento de un sólido rígido . . . . .	130
A.2. Sistema de posiciones SP . . . . .	131
A.3. Plano $XZ$ en los sistemas de coordenadas . . . . .	131
A.4. Centro de gravedad de los componentes principales . . . . .	134
A.5. Propiedades atmosféricas de acuerdo a la ISA . . . . .	136

# Índice de tablas

1.1.	Comparativa de simuladores comerciales . . . . .	19
3.1.	Resumen del orden y precisión de los métodos numéricos utilizados . . . . .	63
3.2.	Matriz de decisión Caso 1 . . . . .	70
3.3.	Matriz de decisión Caso 2 . . . . .	72
3.4.	Matriz de decisión Caso 3 . . . . .	75
3.5.	Resumen de propiedades de los métodos numéricos utilizados . . . . .	76
3.6.	Aplicación recomendada de los métodos numéricos utilizados . . . . .	76
4.1.	Figura estándar que representa a cada componente . . . . .	87
5.1.	Tabla de masas y dimensiones de los componentes principales del cohete Xitle. . . . .	96
5.2.	Tabla de masas y dimensiones de los componentes internos del cohete Xitle. . . . .	97
5.3.	Propiedades geométricas y aerodinámicas de los componentes del cohete Xitle. . . . .	100
5.4.	Resultados clave del vuelo experimental, simulado y de OpenRocket . . . . .	111
5.5.	Errores relativos respecto al vuelo experimental . . . . .	112
5.6.	Comparación del centro de gravedad (CG) y centro de presión (CP) entre el modelo original y el modificado. . . . .	114
5.7.	Comparación de resultados clave con y sin viento . . . . .	117
B.1.	Tabla de Butcher (coeficientes $c_i$ y $a_{ij}$ ) del método Dormand-Prince 8(5,3) . . . . .	138



# A Física de la dinámica de vuelo

**Sólido rígido:** Es un caso especial de un sistema de muchas partículas, y considera que la distancia entre las partículas de estos cuerpos permanece constante ( $R = \text{constante}$ ), i.e. son absolutamente indeformables.

## A.0.1. Sistemas de coordenadas

El movimiento de un objeto, en este caso del cohete, requiere considerarse desde algún punto en específico para ser preciso. Así, la cinemática del cuerpo rígido queda determinada a partir de dos sistemas de referencia, uno móvil fijo al cuerpo y el segundo fijo en el piso.

Para la formulación de las ecuaciones, serán implementadas las coordenadas cartesianas, que requieren tres coordenadas para determinar la posición de un cuerpo rígido.

Como se observa en la figura A.1, la posición  $r_P$  de un punto  $P$  del sólido puede ponerse en función del centro de masa del sólido  $CG$  como:  $r_P = r_{CG} + \vec{P}$ , donde  $r_{CG}$  es el vector de posición del centro de masas del sólido y  $\vec{P}$  el vector que va del centro de masas al punto  $P$ .

### Sistema Fijo (SF)

El sistema, que llamaremos “SF”, es fijo en el plano de cualquier observador que se encuentre en Tierra con origen en el lugar de lanzamiento, donde se posicione la torre de lanzamiento, con latitud y longitud dadas y que no se mueve o rota.

Este sistema se usa para conocer el movimiento general del cohete respecto a los observadores en Tierra y conocer así el posible lugar de caída.

Es importante mencionar que no se toma como marco de referencia el “sistema geocéntrico rotacional ecuatorial” que gira con respecto a la esfera celeste debido a que los cohetes que se consideran en este trabajo no alcanzan altitudes tan grandes, mayores a 50 km, como para verse afectados por el movimiento rotacional de la Tierra.

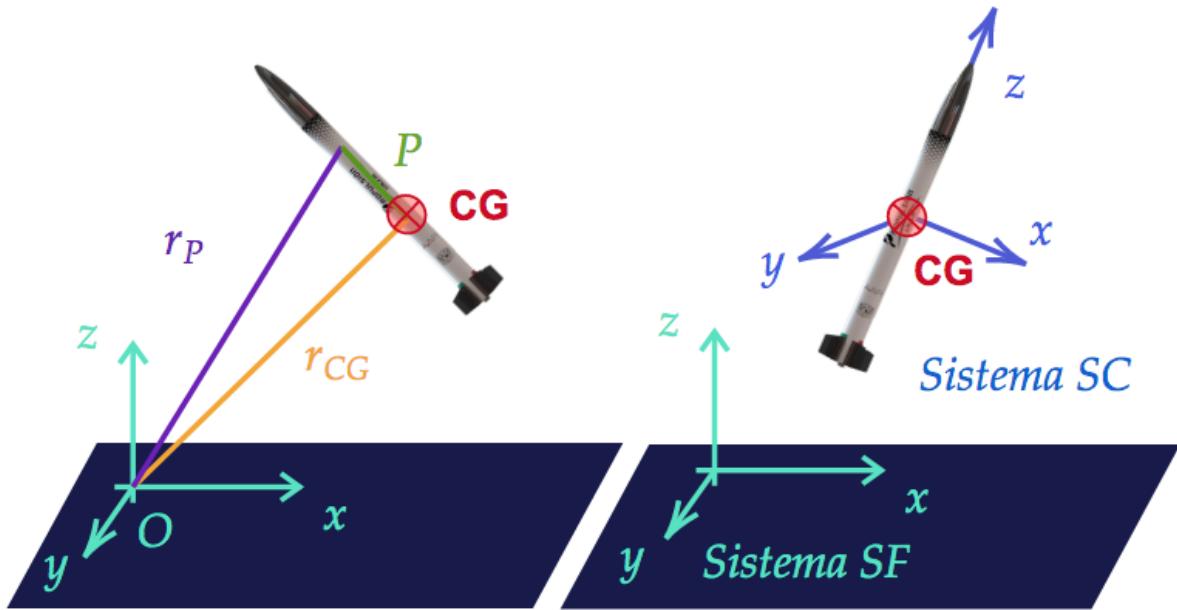


Figura A.1: Representación general del movimiento de un sólido rígido

### Sistema del Cuerpo cohete (SC)

El sistema no inercial es aquel que sigue el movimiento del cohete con origen en su centro de gravedad y cuyo eje  $z$  es aquel que apunta hacia la punta de la nariz. Por otro lado, el eje  $x$  y el eje  $y$  definen el plano tangente al eje  $z$ , como se muestra en la Figura A.1. Notemos que el eje  $x$  lo definimos de tal forma que genera (junto con el eje  $z$ ) al plano de vuelo. Sistema al cualaremos referencia como “SC”.

Este sistema se usa para localizar la aplicación de las fuerzas que afectan al objeto, así como conocer la dirección de su velocidad y aceleración. Estos ejes coinciden con los movimientos de *roll*, *pitch* y *yaw*.

### Sistema de posiciones en cohete (SP)

Este sistema de referencia sigue considerando el eje transversal del cohete como el eje  $z$ . Sin embargo, el origen no se encuentra en el centro de gravedad sino en la punta de la nariz con dirección positiva  $z$  apuntando hacia la base, como se observa en A.2.

Este sistema se usa para indicar la posición de los componentes en el objeto general, que es el cohete.

### Transformación de coordenadas

En este trabajo, se utilizan matrices de rotación como alternativa a los ángulos de Euler. Esta metodología puede simplificarse con el uso de cuaterniones, que además



Figura A.2: Sistema de posiciones SP

no permite singularidades. Sin embargo, en este trabajo no se aborda dicho enfoque. Para transformar un vector  $\vec{v}$  de un sistema de coordenadas a otro, podemos utilizar la siguiente ecuación:

$$v_{destino} = R_{destino-origen} * T_{destino-origen} * v_{origen}$$

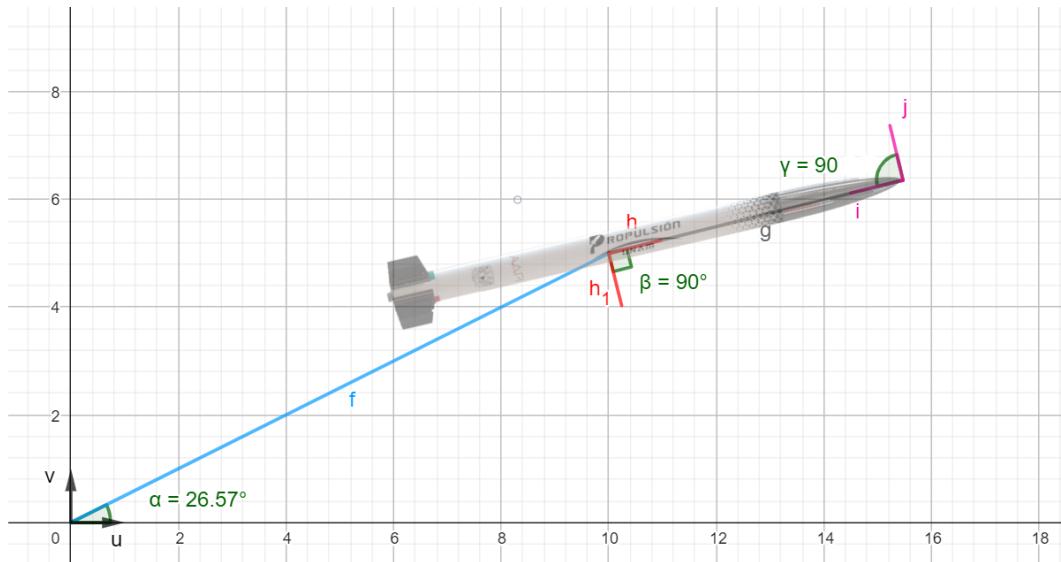


Figura A.3: Plano  $XZ$  en los sistemas de coordenadas

**Matriz de transformación homogénea** Se hace uso de matrices de transformaciones homogéneas, pues simplifica la representación y el cálculo de transformaciones geométricas complejas. Por otro lado, facilita la combinación de transformaciones sucesivas, ya que la multiplicación de matrices homogéneas es asociativa y mantiene la información de escala durante las transformaciones.

Al combinar las matrices de rotación y traslación en una matriz de transformación homogénea ( $H$ ), se logra realizar las transformaciones a las coordenadas del sistema  $SP$ .

**NOTA:** Las transformaciones entre sistemas en el código solo consideran la rotación, la translación se deja de lado, pues la suma y resta de vectores es invariante ante translaciones.

### A.0.2. Cinemática

La cinemática representa el movimiento sin considerar las causas del mismo. Para esto, se necesita conocer la distribución de masa de nuestro objeto.

#### Distribución de masa

La distribución espacial de la masa dentro de un cuerpo sólido determina la manera en que tiende a moverse. Sabemos que el movimiento en un espacio tridimensional, conocido como  $\mathbb{R}^3$ , está caracterizado por la matriz de inercia, cuyas entradas indican los momentos de inercia según tres ejes perpendiculares y tres productos de inercia.

$$I = \begin{bmatrix} I_{xx} & I_{yx} & I_{zx} \\ I_{xy} & I_{yy} & I_{zy} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix}$$

Sin embargo, si el objeto solo rota sobre el eje y, podemos suponer que la velocidad angular solo tiene componente en la *dirección y*, es decir:

$$\vec{\omega} = (0, \omega_y, 0)$$

En este caso, la matriz de inercia rotacional se reduce a:

$$I = \begin{bmatrix} I_{yy} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{yy} \end{bmatrix}$$

donde  $I_{yy}$  es el momento de inercia del objeto con respecto al eje y, conocido como el momento de inercia longitudinal.

En nuestro caso, el cohete tiene permitido moverse sobre el plano,  $XZ$  de modo que solo existen rotaciones sobre el eje  $\vec{y}$ . En consecuencia, únicamente necesitamos obtener el valor del momento de inercia longitudinal.

#### Momento de inercia

El momento de inercia proporciona una medida de la resistencia del sistema cuando se trata de ponerlo en un movimiento de rotación. En el sistema internacional de unidades, el momento de inercia se expresa en  $kg * m^2$ .

El tensor de inercia  $I$  en el sistema de referencia del cuerpo es una matriz simétrica que relaciona el momento angular  $L$  con la velocidad angular  $\omega$  mediante  $\mathbf{L} = \mathbf{I}\omega$

Debido a la axisimetría del cuerpo,  $I_x = I_y$  el momento de inercia longitudinal y  $I_z$  el momento de inercia rotacional. Para este trabajo se utilizan las fórmulas estándar para calcular los valores de  $I_x$  de cada componente. Mientras que,  $I_z = 0$  dado que el “*roll moment*” se consideró nulo durante el vuelo.

Luego, para el momento de inercia general se usó el teorema de Steiner para trasladar los momentos de inercia locales por componente a uno general del cohete. **Teorema de Steiner**

$$I_{\text{trasladado}} = I_{\text{componente}} + \text{masa} * (CG - r)$$

Una vez trasladados se obtuvo el valor general sumando los momentos de inercia individuales.

$$I_{\text{general}} = \sum I_{\text{transladados}}$$

## Cálculo del Centro de gravedad (CG)

El peso  $W$  de un cuerpo es la resultante de las fuerzas másicas distribuidas que la Tierra ejerce sobre los puntos materiales que constituyen el cuerpo. El punto en el que actúa el peso total  $W$  es el centro de gravedad del cuerpo.

Sabemos que el valor de la gravedad no es constante, sino que depende de la distancia del cuerpo al centro de la Tierra. Sin embargo, el tamaño de los cohetes a analizar es despreciable frente al radio de la Tierra y puede suponerse que todos los puntos del cuerpo se hallan a igual distancia del centro de la Tierra, es decir, experimentan la misma atracción gravitatoria. Si además consideramos a la Tierra como plana, esta fuerza será siempre perpendicular, es decir, permanecerá aplicada sobre el eje  $z$ .

De lo anterior, podemos decir que el centro de masa coincide con el centro de gravedad del cuerpo. Por la característica asimétrica del cuerpo del cohete, suponemos que la distribución de masa de cada componente individual es uniforme, i.e. su densidad es constante, aunque esto no sea necesariamente cierto, resulta para nosotros una aproximación adecuada.

De modo que, se hace uso de las fórmulas estándar [Kleppner and Kolenkow, 2014] para el cálculo de los centroides de las formas cónicas o parabólicas y cilíndricas de los componentes, usando el *sistema coordenado SP* recorrido sobre el eje  $z$  a la punta de cada componente. Resultando en el cálculo de cada componente, como se muestra en la Figura A.4.

Cabe mencionar que debido a que el cohete es axisimétrico, la coordenada  $x$  y la coordenada  $y$  del centro de gravedad es cero. Entonces, las coordenadas  $z$  del centro de masa de cada componente están dadas por el centro geométrico en el plano  $XZ$ .

Finalmente, se combinan en un centro de gravedad general del cuerpo de la siguiente manera:

$$CG = \sum m_i CG_i, i \in \{1, 2, \dots, n\},$$

donde  $n$  la cantidad total de componentes que conforman el vehículo y  $m_i$  y  $CG_i$  representan la masa y el centro de gravedad de cada componente.



Figura A.4: Centro de gravedad de los componentes principales

### Cálculo del Centro de presión (CP)

Existen diversos métodos para el cálculo del centro de presión de vehículos delgados con aletas, cohetes o misiles. Sin embargo, el método más común es el desarrollado por James Barrowman en su tesis de maestría [Barrowman, 1966].

El método de Barrowman calcula las principales características aerodinámicas: los coeficientes de fuerza normal, centros de presiones de los componentes y del cohete completo, derivada del coeficiente de momento forzado del ruedo (en inglés “*roll forcing moment coefficient derivative*”), derivada del coeficiente de momento de amortiguación del balanceo (en inglés, “*roll damping moment coefficient derivative*”), la derivada del coeficiente de momento de amortiguación del cabeceo (en inglés, “*pitch damping moment coefficient derivative*”) y el coeficiente de arrastre.

Para calcular las derivadas de los coeficientes de fuerza normal de la nariz, aletas y *boattail* se utilizan las siguientes fórmulas:

$$CN\alpha_{nariz} = 2.2$$

$$CN\alpha_{aletas} = \frac{4n \left(\frac{s}{d}\right)^2}{1 + \sqrt{1 + \left(\frac{2L_F}{C_R + C_T}\right)^2}}$$

$$CN\alpha_{boattail} = -2 \left(1 - \left(\frac{d_R}{d}\right)^2\right)$$

Para el cálculo de los centros de presiones se realiza lo siguiente:

$$CP_{nariz} = \frac{2}{3} * L(\text{cónica})$$

$$CP_{nariz} = 0.466 * L(\text{ojiva tangente})$$

$$CP_{nariz} = \frac{1}{2} * L(\text{paraboloide})$$

$$CP_{nariz} = \frac{1}{3} * L(\text{elipsoidal})$$

$$CP_{aletas} = X_B + \frac{X_R}{3} \left( \frac{C_R + 2C_T}{C_R + C_T} \right) + \frac{1}{6} (C_R + C_T - \frac{C_R C_T}{C_R + C_T})$$

$$CP_{transicin} = X_P + \frac{L_T}{3} \left( 1 + \frac{1 - \frac{d}{d_R}}{1 - (\frac{d}{d_R})^2} \right),$$

donde  $CL$  es (falta explicar las variables de las ecuaciones)

De esta forma, el coeficiente normal y centro de presión de todo el vehículo se calculan de la siguiente manera:

$$CN_\alpha = CN\alpha_{nariz} + CN\alpha_{aletas} + CN\alpha_{transicin} \quad (\text{A.1})$$

$$CP = \frac{CN\alpha_{nariz} * CP_{nariz} + CN\alpha_{aletas} * CP_{aletas} + CN\alpha_{transicin} * CP_{transicin}}{CN_\alpha} \quad (\text{A.2})$$

## A.1. Modelo atmosférico

La ISA (Atmósfera Estándar Internacional) es un modelo que describe la distribución vertical de la temperatura, la presión, la densidad y otras propiedades atmosféricas en las distintas capas de la atmósfera de manera promedio en condiciones atmosféricas típicas, tomando como referencia el nivel del mar.

En este trabajo, las propiedades atmosféricas como la densidad del aire, presión, temperatura, velocidad del sonido se calculan para cada altura de acuerdo con las fórmulas encontradas en [isa, 1971]

En este modelo se tienen en cuenta las ecuaciones de gases ideales y el equilibrio de fuerzas. En general, cada capa está determinada por valores iniciales y su variación; si el coeficiente adiabático es distinto de cero, se calcula de la siguiente forma:

$$\Delta T(z) = T_0 + z \quad (\text{A.3})$$

$$\Delta P(z) = P_0(1 + zT_0) - gR \quad (\text{A.4})$$

$$\Delta \rho(z) = 0(1 + zT_0) - gR - 1 \quad (\text{A.5})$$

En algunas capas la temperatura disminuye linealmente con la altura. Mientras que a medida que la altitud aumenta, la presión y la densidad disminuyen exponencialmente, como se muestra en la Figura A.5.

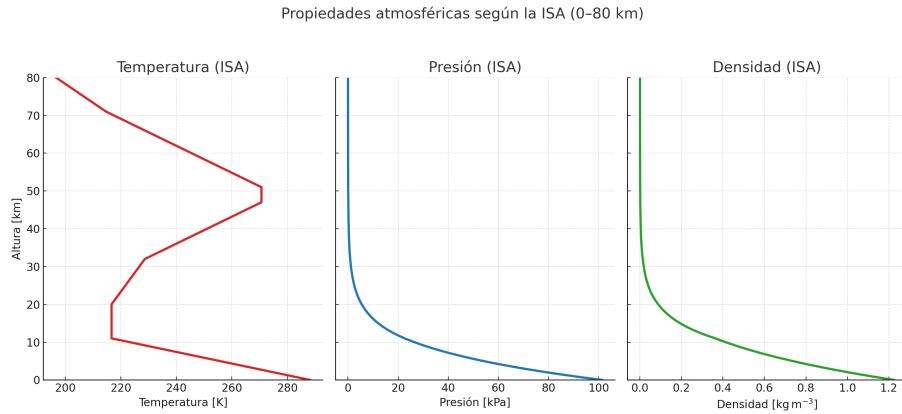


Figura A.5: Propiedades atmosféricas de acuerdo a la ISA

Es importante tener en cuenta que la ISA es un modelo idealizado y la atmósfera real puede desviarse de este modelo debido a varias influencias, incluyendo factores climáticos y geográficos.

# B Método de Dormand-Prince 8(5,3)

El método Dormand-Prince 8(5,3), comúnmente abreviado como DOPRI853, es un esquema explícito adaptativo de la familia de Runge-Kutta embebidos. Está diseñado para obtener una solución de alto orden (orden 8), junto con soluciones embebidas de orden 5 y 3 que permiten estimar el error local y controlar el tamaño del paso de forma robusta. Este método es altamente eficiente en simulaciones de larga duración que requieren precisión elevada en sistemas no rígidos.

## B.1. Esquema general

Dado el problema de valor inicial

$$\frac{dy(t)}{dt} = f(t, y(t)), \quad y(t_0) = y_0, \quad (\text{B.1})$$

el método calcula una serie de etapas intermedias  $k_i$  de la forma:

$$k_i = hf \left( t_n + c_i h, y_n + \sum_{j=1}^{i-1} a_{ij} k_j \right), \quad (\text{B.2})$$

y luego combina estas etapas con los pesos  $b_i$ ,  $\hat{b}_i$ , y  $d_i$  para obtener estimaciones de distintas órdenes.

## B.2. Tabla de Butcher

La siguiente tabla (Tabla B.1) resume los coeficientes  $c_i$  y  $a_{ij}$  del esquema de Butcher utilizado en el método DOPRI853. Se omiten ceros explícitos para mayor claridad.

Tabla B.1: Tabla de Butcher (coeficientes  $c_i$  y  $a_{ij}$ ) del método Dormand-Prince 8(5,3)

$c_i$	$a_{i1}$	$a_{i2}$	$a_{i3}$	$a_{i4}$	$a_{i5}$	$a_{i6}$	$a_{i7}$	$a_{i8}$	$a_{i9}$	$a_{i10}$	$a_{i11}$
0.00000											
0.05556	0.05556										
0.08333	0.02083	0.06250									
0.12500	0.03125		0.09375								
0.31250	0.00434		0.04219	0.32552							
0.37500	0.02344			0.18984	0.67500						
0.14750	0.00521				0.11391	0.35048	0.10000				
0.46500	0.00391				0.08760	0.35048	0.08438	0.07500			
0.72900	0.00156				0.02921	0.11683	0.12656	0.22500	0.09375		
1.00000	0.00313				0.04219	0.16875	0.28125	0.15000	0.11250	0.33750	
1.00000	0.00417				0.04821	0.19286	0.32143	0.16667	0.12500	0.37500	0.10000
1.00000	0.00417				0.04821	0.19286	0.32143	0.16667	0.12500	0.37500	0.10000

### B.3. Pesos y estimaciones embebidas

La solución de orden 8 se calcula con:

$$y_{n+1}^{(8)} = y_n + \sum_{i=1}^{12} b_i k_i, \quad (\text{B.3})$$

donde:

$$\begin{aligned} b_1 &= \frac{14005451}{335480064}, & b_2 &= 0, \\ b_3 &= 0, & b_4 &= 0, \\ b_5 &= 0, & b_6 &= \frac{-59238493}{1068277825}, \\ b_7 &= \frac{181606767}{758867731}, & b_8 &= \frac{561292985}{797845732}, \\ b_9 &= \frac{-1041891430}{1371343529}, & b_{10} &= \frac{760417239}{1151165299}, \\ b_{11} &= \frac{118820643}{751138087}, & b_{12} &= \frac{-528747749}{2220607170} \end{aligned}$$

La solución embebida de orden 5 es:

$$y_{n+1}^{(5)} = y_n + \sum_{i=1}^{12} \hat{b}_i k_i, \quad (\text{B.4})$$

con:

$$\begin{aligned}
\hat{b}_1 &= \frac{13451932}{455176623}, & \hat{b}_2 &= 0, \\
\hat{b}_3 &= 0, & \hat{b}_4 &= 0, \\
\hat{b}_5 &= 0, & \hat{b}_6 &= \frac{-808719846}{976000145}, \\
\hat{b}_7 &= \frac{1757004468}{5645159321}, & \hat{b}_8 &= \frac{656045339}{265891186}, \\
\hat{b}_9 &= \frac{-3867574721}{1518517206}, & \hat{b}_{10} &= \frac{465885868}{322736535}, \\
\hat{b}_{11} &= \frac{53011238}{667516719}, & \hat{b}_{12} &= \frac{2}{45}
\end{aligned}$$

Finalmente, la solución de orden 3:

$$y_{n+1}^{(3)} = y_n + \sum_{i=1}^{12} d_i k_i, \quad (\text{B.5})$$

con:

$$\begin{aligned}
d_1 &= \frac{10425724}{595395017}, & d_2 &= 0, \\
d_3 &= 0, & d_4 &= 0, \\
d_5 &= 0, & d_6 &= 0, \\
d_7 &= \frac{137134352}{785453755}, & d_8 &= \frac{65450036}{464620123}, \\
d_9 &= \frac{25170878}{318867523}, & d_{10} &= \frac{35692889}{335116620}, \\
d_{11} &= 0, & d_{12} &= 0
\end{aligned}$$



# C Repositorio de código en Github

## C.1. Instrucciones de uso del simulador

El código fuente del simulador de trayectorias desarrollado como parte de este trabajo se encuentra disponible en el siguiente repositorio de GitHub:

[github.com/Meijix/RocketTrajectorySimulator-PropulsionUnam](https://github.com/Meijix/RocketTrajectorySimulator-PropulsionUnam)

### C.1.1. Requisitos del sistema

Para ejecutar el simulador se requiere contar con:

- Python 3.8 o superior
- Sistema operativo compatible (Windows, Linux o macOS)
- Acceso a un entorno con permisos de instalación de paquetes

### C.1.2. Instalación de dependencias

Una vez clonado el repositorio, es recomendable crear un entorno virtual e instalar las dependencias necesarias. Para ello, desde la terminal:

```
git clone https://github.com/Meijix/RocketTrajectorySimulator-PropulsionUnam
cd RocketTrajectorySimulator-PropulsionUnam
python -m venv env
source env/bin/activate # en Windows: env\Scripts\activate
pip install -r requirements.txt
```

### C.1.3. Ejecución del simulador

Para ejecutar la interfaz gráfica del simulador:

```
python main_gui.py
```

Para correr simulaciones específicas sin GUI, se pueden ejecutar los scripts directamente dentro de la carpeta **Simulador/Pruebas/** modificando los parámetros deseados.

#### C.1.4. Estructura general del repositorio

- **Simulador/** – Núcleo del simulador y scripts de integración.
- **GUI/** – Código de la interfaz gráfica basada en Tkinter.
- **Paquetes/** – Funciones auxiliares (viento, fuerzas, condiciones).
- **Archivos/** – Archivos de entrada como curvas de empuje y masa.
- **Resultados/** – Carpeta sugerida para guardar salidas de simulación.

#### C.1.5. Licencia y colaboración

El proyecto se encuentra disponible bajo una licencia de código abierto (MIT). Se invita a estudiantes, investigadores y equipos de cohetería experimental a contribuir con mejoras, reportes de errores o sugerencias a través del repositorio oficial.