



UNIVERSIDAD  
POLITÉCNICA  
DE MADRID

---

# TRABAJO 3

## DINÁMICA DE LA REENTRADA

AERODINÁMICA DE ALTAS VELOCIDADES  
Y  
FENÓMENOS DE REENTRADA

---

*Autor:* Daniel DEL RÍO VELILLA

*Profesor:* Sebastián FRANCHINI

MADRID, 28 DE MAYO DE 2021

*Repositorio del proyecto* [https://github.com/Danieldelriovelilla/AAV\\_Trabajos.git](https://github.com/Danieldelriovelilla/AAV_Trabajos.git)

# Índice

<b>Índice de figuras</b>	<b>I</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Solución numérica</b>	<b>1</b>
2.1. Ecuaciones . . . . .	1
2.2. Código . . . . .	2
2.3. Resultados . . . . .	5
<b>3. Validación de resultados numéricos</b>	<b>9</b>
3.1. Ecuaciones analíticas . . . . .	9
3.2. Código . . . . .	10
3.3. Comparación de resultados . . . . .	12
<b>A. Código</b>	<b>16</b>
A.1. Programa principal . . . . .	16
<b>Referencias</b>	<b>23</b>

## Índice de figuras

1.	Evolución temporal de las variables de interés para unas condiciones iniciales de $u_e = 7,5$ [km/s] y $\gamma_e = -5$ [ $^{\circ}$ ]. . . . .	6
2.	Evolución temporal de las variables de interés para unas condiciones iniciales de $u_e = 7,5$ [km/s] y $\gamma_e = -10$ [ $^{\circ}$ ]. . . . .	7
3.	Evolución temporal de las variables de interés para unas condiciones iniciales de $u_e = 8,5$ [km/s] y $\gamma_e = -5$ [ $^{\circ}$ ]. . . . .	8
4.	Trayectoria cápsula Apolo [1] . . . . .	9
5.	Comparación de resultados en la entrada balística entre el método numérico y analítico. . . . .	12
6.	Comparación de resultados en la entrada sustentada entre el método numérico y analítico. . . . .	14
7.	Sustentación y resistencia aerodinámica a lo largo de la reentrada para la segunda combinación de condiciones iniciales y $\beta = 50$ . . . . .	15

## 1. Introducción

En este documento se presenta el desarrollo del trabajo número 3 correspondiente a la parte teórica de la asignatura Aerodinámica de Altas Velocidades y Fenómenos de Reentrada.

En él se expondrán las ecuaciones que rigen la *Dinámica de la Reentrada*, después se pasará a comentar la implementación de estas ecuaciones en MATLAB. Posteriormente se mostrarán los resultados obtenidos para varias condiciones iniciales de la reentrada. Finalmente se compararán estos resultados con los que aportan las soluciones analíticas y se comentarán sus diferencias.

El programa principal se encuentra en el anexo A.1. En este programa se integran las ecuaciones diferenciales y evalúan las ecuaciones analíticas y se pintan los resultados. En este programa principal se definen también las variables globales que se utilizarán en las sucesivas funciones.

## 2. Solución numérica

Las ecuaciones que se muestran a continuación se han extraído de [1].

### 2.1. Ecuaciones

Las ecuaciones diferenciales que rigen la dinámica de la reentrada son las siguientes,

$$\frac{d\gamma}{dt} = \frac{1}{u} \left( L(z, u) - g(z) \cdot \cos \gamma + \frac{u^2}{r} \cos(\gamma) \right), \quad (1)$$

$$\frac{du}{dt} = -g(z) \cdot \sin \gamma - D(z, u), \quad (2)$$

$$\frac{dz}{dt} = u \cdot \sin \gamma. \quad (3)$$

Donde las diversas funciones que aparecen en los términos de la derecha de las ecuaciones diferenciales son las fuerzas aerodinámicas,

$$L(z, u) = \frac{\rho(z)u^2 E}{2\beta},$$

$$D(z, u) = \frac{\rho(z)u^2}{2\beta},$$

siendo la definición del coeficiente balístico

$$\beta = \frac{m}{Sc_D},$$

y de la eficiencia aerodinámica

$$E = \frac{c_L}{c_D} .$$

La gravedad,

$$g_0 = 9,81 \left[ \frac{\text{m}}{\text{s}^2} \right],$$

$$g(z) = g_0 \left( \frac{R_T}{z + R_T} \right)^2 .$$

y la densidad,

$$\begin{cases} \rho_0 = 1,225 \left[ \frac{\text{kg}}{\text{m}^3} \right] & \& z_s = 7524 \text{ [m]} / z \leq 150 \cdot 10^3 \text{ [m]} \\ \rho_0 = 3,875 \cdot 10^{-9} \left[ \frac{\text{kg}}{\text{m}^3} \right] & \& z_s = 59060 \text{ [m]} / z > 150 \cdot 10^3 \text{ [m]} \end{cases},$$

$$\rho(z) = \rho_0 \exp\left(-\frac{z}{z_s}\right) \left[ \frac{\text{kg}}{\text{m}^3} \right].$$

Además, se relaciona la altura y el radio mediante la siguiente expresión

$$R_T = 6378 \cdot 10^3 \text{ [m]},$$

$$r = z + R_T.$$

## 2.2. Código

La realización del programa se ha orientado desde un punto de vista funcional, por lo que a continuación se van a presentar las diversas funciones que realizan la integración numérica del sistema de ecuaciones diferenciales para diversas condiciones iniciales. La altura de entrada para las tres condiciones será igual a 100 km, a partir de la cual se considera la atmósfera como un medio continuo, la primera combinación se tomará como referencia y después se variará la velocidad al mismo ángulo inicial y finalmente ángulo distinto para la misma velocidad de referencia.

- ***Modelo de gravedad***

Esta función tiene como argumento de entrada un vector de alturas y devuelve otro vector que contiene el valor de la gravedad para cada una de las alturas.

```

1 function g = Function_g(z)
2
3 % Global variables
4 global g0;
5 global Rt;
```

```
6
7      % Function body
8      g = g0*( Rt./(z + Rt) ).^2;
9
10 end
```

- **Modelo de atmósfera**

Esta función tiene como argumento de entrada un vector de alturas y como argumentos de salida la densidad para cada altura y otros dos vectores de las constantes  $z_s$  y  $\rho_0$  correspondientes.

```
1 function [rho, zs, rho0] = Function_rho(z)
2
3     % Function body
4     zs = ones(size(z))*59.06e3;
5     zs(z<=150e3) = 7.524e3;
6
7     rho0 = ones(size(z))*3.875e-9;           % [kg/m^3]
8     rho0(z<=150e3) = 1.225;                 % [kg/m^3]
9
10    rho = rho0.*exp(-z./zs);
11
12 end
```

- **Sistema de ecuaciones diferenciales**

Para que el esquema numérico pueda realizar el trabajo eficientemente se han introducido las ecuaciones diferenciales en una función independiente, que devuelve el valor de las derivadas teniendo como entradas los valores de las variables y las características de la cápsula.

En ella se utilizan las funciones de gravedad y densidad definidas anteriormente.

```
1 function dydt = Ec_Dinamica(y, E, beta)
2
3     % Global variables
4     global Rt;
5
6     % Parameters
7     gamma = y(1);
8     u = y(2);
9     z = y(3);
10
11    % z to r switch
12    r = z + Rt;
13
```

```

14     % Density and gravity
15     [rho, _, _] = Function_rho(z);
16     g = Function_g(z);
17
18     % Aero forces
19     Lom = rho * u^2 * E / (2*beta);
20     Dom = rho * u^2 / (2*beta);
21
22     % Differential equations
23     gamaP = 1/u*( Lom - g*cos(gamma) + u^2/r*cos(gamma) );
24     uP = -g*sin(gamma) - Dom;
25     zP = u*sin(gamma);
26     dydt = [gamaP; uP; zP];
27
28 end

```

- ***Esquema numérico de integración***

Finalmente se ha implementado el integrador a utilizar. El esquema numérico elegido ha sido un Runge Kutta de orden 4. Para construir esta función se ha utilizado los programas de referencia en el moodle de la asignatura y la información obtenida en [2].

Esta función tiene como argumento de entrada las condiciones iniciales, junto a las características de la cáspula e integra las ecuaciones definidas en la función *Ec\_Dinamica* hasta que la altura es igual a 0 y devuelve un *Struct* con la solución.

```

1 function [entryDinamicSol] = Runge_Kutta_4(beta, E, yi, Dt)
2
3     % Initial conditions
4     y = [yi(3); yi(2); yi(1)];
5     mov = [y; 0];
6
7     % Integrate while z is positive
8     while y(3) > 0
9         k1 = Ec_Dinamica(y, E, beta);
10        k2 = Ec_Dinamica(y + k1*Dt/2, E, beta);
11        k3 = Ec_Dinamica(y + k2*Dt/2, E, beta);
12        k4 = Ec_Dinamica(y + k3*Dt, E, beta);
13
14        ynl = y + Dt/6*(k1 + 2*k2 + 2*k3 + k4);
15        aux = [ynl; k1(2)];
16        mov = [mov aux];
17
18        y = ynl;
19    end

```

```
20     mov = mov(:,1:end-1);
21
22     t      = (0:length(mov)-1)*Dt;
23     gamat = mov(1,:);
24     ut    = mov(2,:);
25     zt    = mov(3,:);
26     at    = mov(4,:);
27
28 % Load the solution in a struct
29 entryDinamicSol.z0      = yi(1);
30 entryDinamicSol.u0      = yi(2);
31 entryDinamicSol.gama0   = yi(3);
32 entryDinamicSol.beta0   = beta;
33 entryDinamicSol.E0      = E;
34 entryDinamicSol.t = t';
35 entryDinamicSol.tx = max(t);
36 entryDinamicSol.z = zt';
37 entryDinamicSol.zx = max(zt);
38 entryDinamicSol.u = ut';
39 entryDinamicSol.ux = max(ut);
40 entryDinamicSol.gamma = gamat';
41 entryDinamicSol.gammax = max(gamat);
42 entryDinamicSol.a = at';
43 entryDinamicSol.ax = min(at);
44
45 end
```

### 2.3. Resultados

Tras integrar las ecuaciones con tres combinaciones de condiciones iniciales se han representado los resultados obtenidos en las Figuras 1, 2 y 3.

En las figuras se puede ver la evolución de la altura,  $z$  (a), de la velocidad respecto a la velocidad de entrada inicial,  $U/U_e$  (b), del ángulo de descenso,  $\gamma$  (c), y del factor de carga,  $n$  (d), respecto al tiempo.

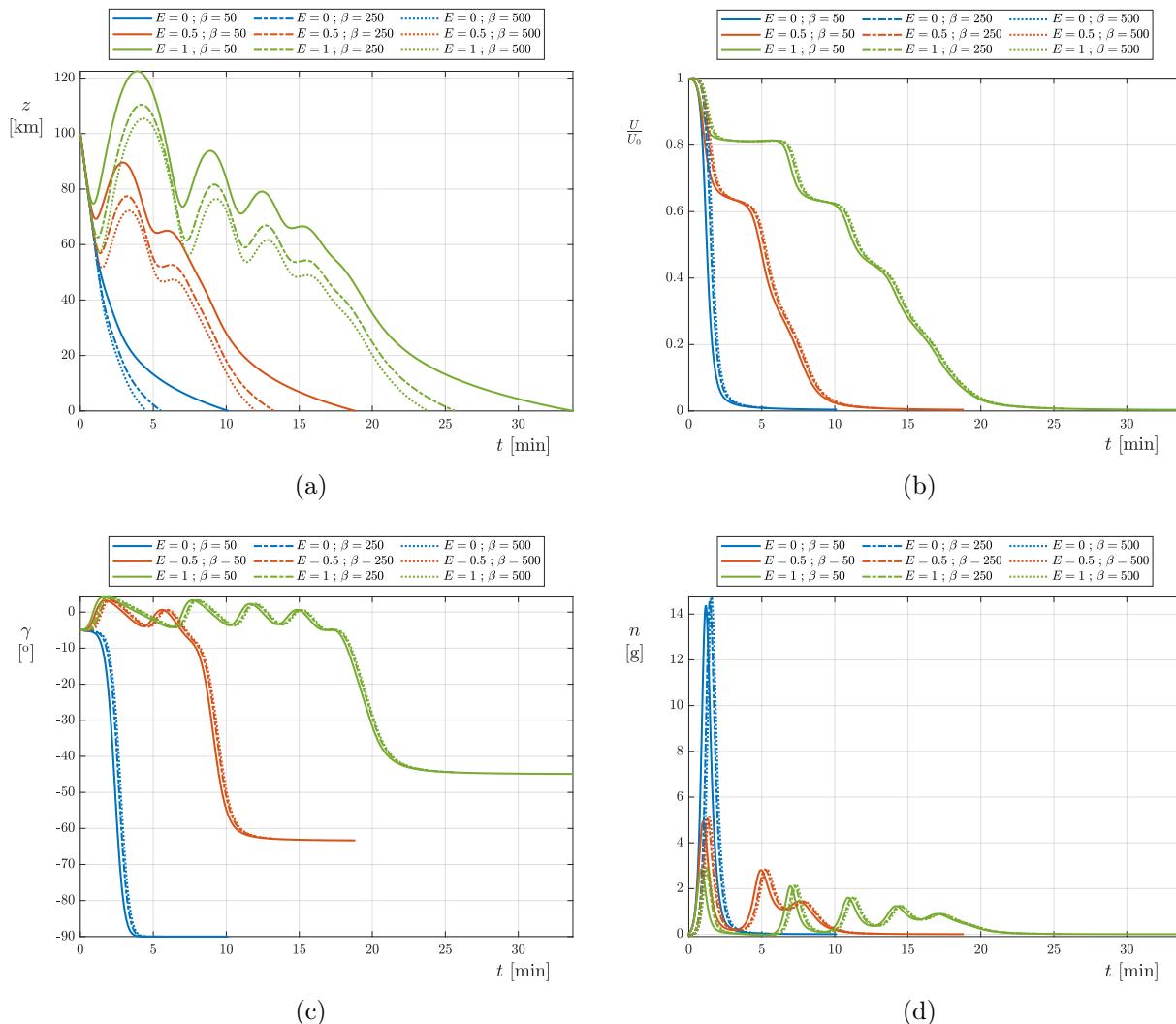


Figura 1: Evolución temporal de las variables de interés para unas condiciones iniciales de  $u_e = 7,5$  [km/s] y  $\gamma_e = -5$  [ $^{\circ}$ ].

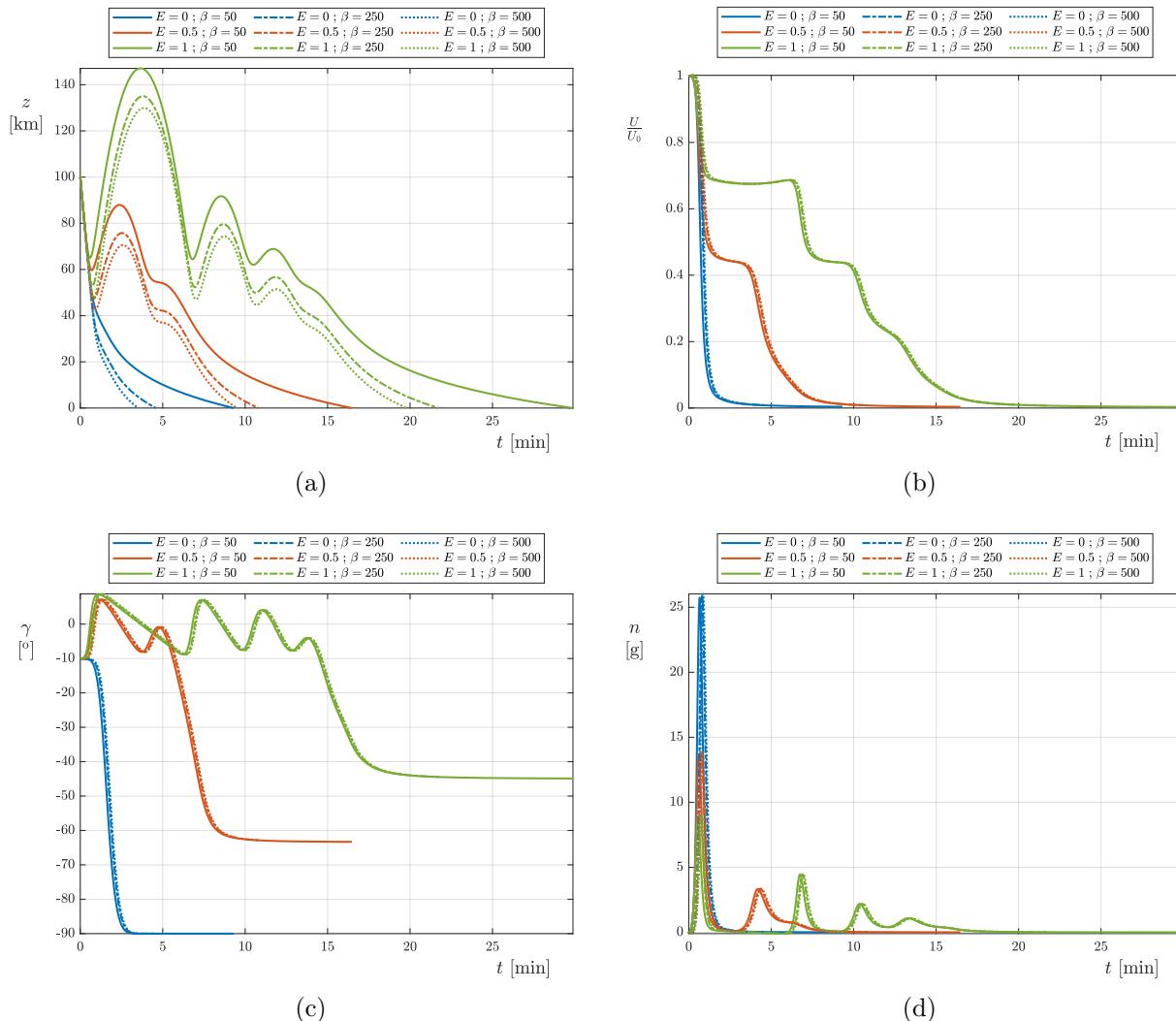


Figura 2: Evolución temporal de las variables de interés para unas condiciones iniciales de  $u_e = 7,5$  [km/s] y  $\gamma_e = -10$  [°].

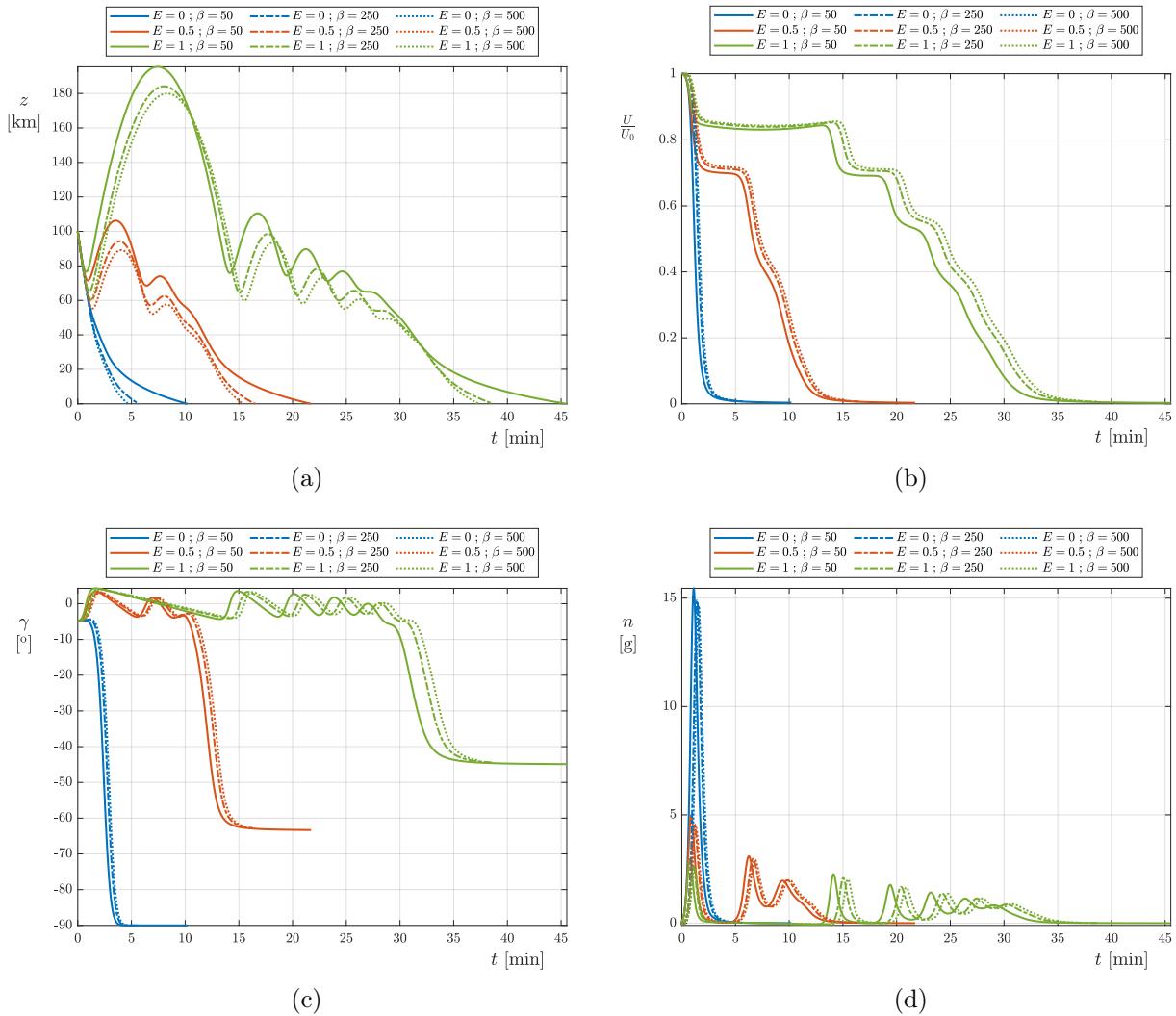


Figura 3: Evolución temporal de las variables de interés para unas condiciones iniciales de  $u_e = 8,5$  [km/s] y  $\gamma_e = -5$  [°].

Se puede comprobar como, independientemente de las condiciones iniciales y del coeficiente balístico, cuando se realiza una reentrada con sustentación ( $E \neq 0$ ) el factor de carga al que se ve sometida la cápsula (Figuras 1d, 2d y 3d) es un orden de magnitud menos que en la entrada balística ( $E = 0$ ). Por lo tanto, si se quiere reentrar una cápsula tripulada siempre se va a tener que realizar con sustentación ya que un humano no puede soportar más de 12 g [1].

Como era de esperar, a mayor eficiencia aerodinámica mayor es la duración de la maniobra. Por otro lado, la relación de velocidades disminuye de forma más paulatina cuanto mayor es la eficiencia (Figuras 1b, 2b y 3b). También se consigue un menor ángulo de descenso cuando la eficiencia es mayor, (Figuras 1c, 2c y 3c).

Como último comentario común a todas las curvas es destacable decir que el coeficiente balístico no tiene una influencia apenas notable comparada con la eficiencia.

Pasando a comparar los resultados entre las diferentes condiciones iniciales se puede afirmar que cuando la maniobra comienza con mayor ángulo inicial su duración es menor, pero a costa de que el factor de carga es hasta dos veces superior que con un ángulo de la mitad de tamaño.

En todas las curvas se puede ver como en el instante en el que la cápsula está sometida al máximo nivel de carga aparece un incremento brusco en la altitud, siendo mayor cuanta mayor es la velocidad de reentrada. Esto puede ser preligero ya que, como se puede ver en la Figura 4, puede aparecer un fenómeno de *rebote* y hacer que la cápsula reorbite.

Estas oscilaciones se analizarán más en profundidad adelante en el documento.

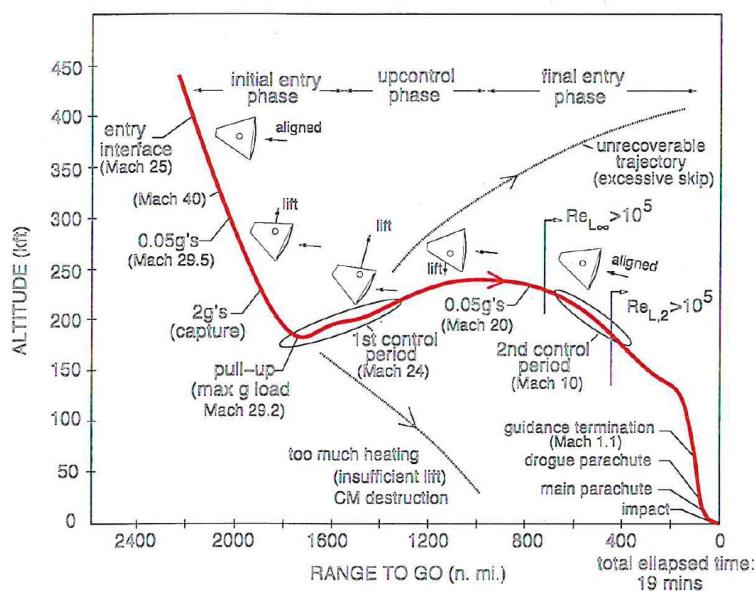


Figura 4: Trayectoria cápsula Apolo [1]

### 3. Validación de resultados numéricos

Una vez obtenidos los resultados numéricos se van a comparar con la solución analítica para así validar los resultados.

#### 3.1. Ecuaciones analíticas

Las ecuaciones analíticas simplificadas se pueden ver a continuación. Todo el proceso de simplificación desde las ecuaciones diferenciales hasta las analíticas se encuentra detallado en [1].

- *Reentrada balística*

$$B = \frac{z_s \rho_0}{2\beta \sin \gamma_e},$$

$$C = \frac{u_e^2 \rho_0}{2\beta g_0},$$

$$\frac{U}{U_e} = \exp \left( B \cdot \exp \left( -\frac{z}{z_e} \right) \right),$$

$$n = C \cdot \exp \left( 2B \cdot \exp \left( -\frac{z}{z_e} \right) - \frac{z}{z_s} \right).$$

- *Reentrada sustentada*

$$\frac{U}{U_e} = \left( \sqrt{1 + \frac{ER_T \rho(z)}{2\beta}} \right)^{-1},$$

$$n = \frac{1}{E} \left[ 1 - \left( \frac{u}{u_e} \right)^2 \right] = \left( \sqrt{E + \frac{2\beta \cdot \exp(-\frac{z}{z_e})}{\rho_0 R_T}} \right)^{-1}.$$

### 3.2. Código

Igual que se ha hecho en el apartado anterior, se han implementado estas ecuaciones en funciones independientes que luego serán llamadas desde el programa principal.

- *Ecuaciones reentrada balística analíticas*

Esta función calcula la relación de velocidades y factor de forma analítica.

```

1 function [U_Ue, n] = Function_Ballistic(z, gamma_e, beta, Ue)
2
3 % Global variables
4 global g0
5
6 % Function body
7 [~, zs, rho0] = Function_rho(z);
8
9 B = zs.*rho0./(2*beta*sin(gamma_e));
10 C = Ue^2.*rho0./(2*beta*g0);
11
12 U_Ue = exp( B.*exp(-z./zs) );
13 n = C.*exp( 2*B.*exp(-z./zs) - z./zs );
14
15 end

```

- **Ecuaciones reentrada sustentada analíticas**

Función que calcula la reentrada sustentada a partir de las ecuaciones analíticas

```
1 function [U_Ue, n] = Function_Glide(z, beta, E)
2
3 % Global variables
4 global Rt
5
6 % Function body
7 [rho, zs, rho0] = Function_rho(z);
8
9 U_Ue = 1./sqrt( 1 + E.*Rt.*rho./( 2*beta ) );
10 n = 1./( E + 2*beta*exp(z./zs) ./(rho0*Rt) );
11
12 end
```

- **Sistema de ecuaciones diferenciales**

Función que ejecuta las dos anteriores y recoge los resultados para facilitar su posterior representación.

```
1 function [ballistic, glide] = Function_Analytic(beta, E, y0)
2
3 % Height array
4 z = linspace(0,y0(1),1e4+1); % [m]
5
6 % Evaluate analytic equations
7 for b = 1:length(beta)
8
9     % Ballistic
10    ballistic(b).beta = beta(b);
11    ballistic(b).z = z;
12    [ballistic(b).u, ballistic(b).n] = Function_Ballistic(z, y0(3), ...
13                beta(b), y0(2));
14
15     % Glide
16    for e = 2:length(E)
17        glide(b,e).beta = beta(b);
18        glide(b,e).E = E(e);
19        glide(b,e).z = z;
20        [glide(b,e).u, glide(b,e).n] = Function_Glide(z, beta(b), E(e));
21    end
22 end
```

23  
24  
25 end

### 3.3. Comparación de resultados

En la Figura 5 se puede ver como en la entrada balística los perfiles que forman la solución numérica y analítica guardan una gran similitud. Esto ocurre para los dos coeficientes balísticos y las dos condiciones iniciales representadas.

Las diferencias que aparecen se deben a las simplificaciones realizadas para llegar al modelo analítico de reentrada.

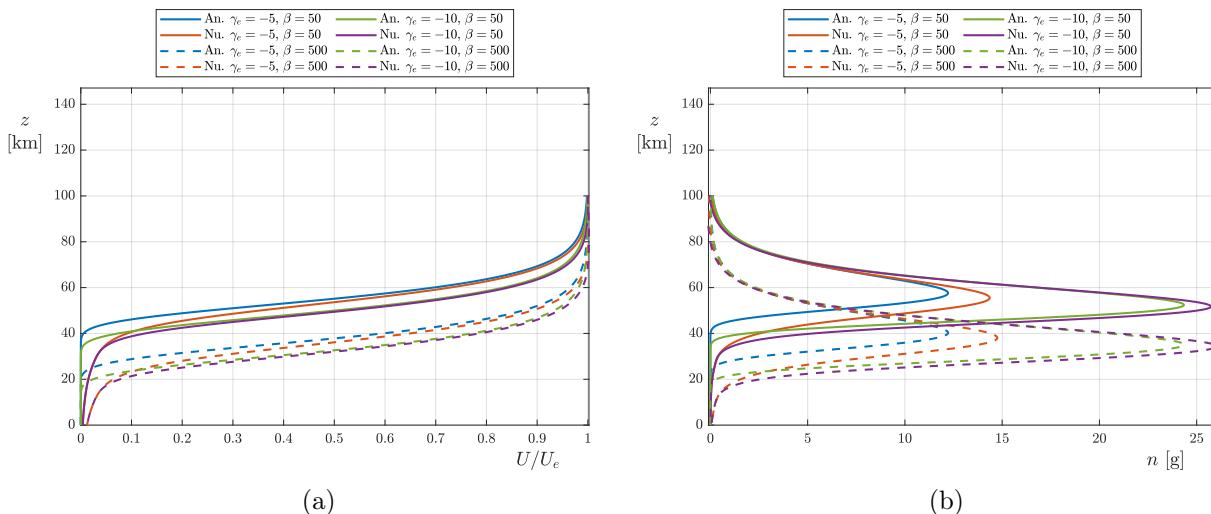


Figura 5: Comparación de resultados en la entrada balística entre el método numérico y analítico.

Para entender la notable diferencia entre el numérico y el analítico para el caso de reentrada con sustentación y condición inicial de velocidad igual a 8,5 km/s, Figura 6, es necesario estudiar la estabilidad del sistema de ecuaciones ante pequeñas perturbaciones.

$$\gamma = \Delta\gamma ,$$

$$z = z_0 + \Delta z .$$

Y si introducimos las perturbaciones en las ecuaciones (1), (2) y (3) queda el siguiente sistema,

$$\frac{L}{m} - g = \Delta \dot{\gamma} u - \frac{u^2}{r}, \quad (4)$$

$$\frac{\Delta \dot{\gamma} u}{g} = \frac{L}{mg} + \frac{u^2}{gr} - 1 = \frac{L}{mg} - \left(1 - \frac{u^2}{u_e^2}\right), \quad (5)$$

$$\Delta \ddot{z} = u \Delta \gamma, \quad (6)$$

(7)

donde se obtiene la aceleración volviendo a derivar 6 respecto al tiempo

$$\Delta \ddot{z} = u \Delta \gamma. \quad (8)$$

Adimensionalizando (8) con la gravedad e igualándola a (5) se obtiene

$$\frac{\Delta \ddot{z}}{g} = \frac{L}{mg} - \left[1 - \left(\frac{u}{u_e}\right)^2\right]. \quad (9)$$

Por otro lado, desarrollando el término de la sustentación y densidad se llega a

$$\frac{L}{mg} = \frac{1}{2} \frac{\rho u^2 Sc_L}{mg} = \frac{1}{2} \frac{\rho_0 u^2 Sc_L}{mg} \exp\left(-\frac{z_0 + \Delta z}{z_s}\right) = \frac{1}{2} \frac{\rho_0 u^2 Sc_L}{mg} \exp\left(-\frac{z_0}{z_s}\right) \exp\left(-\frac{\Delta z}{z_s}\right). \quad (10)$$

Construyendo una sustentación de referencia  $L_0$  con el término de  $z_0$  se obtiene finalmente

$$\frac{L}{mg} = \frac{L_0}{mg} \left(1 - \frac{\Delta z}{z_s}\right) = \frac{L_0}{W} + \frac{\Delta L}{W}. \quad (11)$$

Introduciendo el resultado final de (11) en (9) se llega a

$$\Delta \ddot{z} + \frac{L_0 g}{W z_s} \Delta z = \frac{L_0 g}{W} - g \left[1 - \left(\frac{u}{u_e}\right)^2\right]. \quad (12)$$

que es la ecuación de un oscilador forzado sin amortiguamiento.

Su ecuación homogénea es

$$\Delta \ddot{z} + \omega^2 \Delta z = 0,$$

$$\omega^2 = \frac{L_0 g}{W z_s} = \frac{1}{z_s} \left[1 - \left(\frac{u}{u_e}\right)^2\right].$$

Esta oscilación tiene un periodo igual a

$$T = \frac{2\pi}{\sqrt{\frac{1}{z_s} \left(1 - \frac{u_e^2}{u_e^2}\right)}} \quad [\text{s}] . \quad (13)$$

Este fenómeno generado por perturbaciones en la sustentación se llama *Fugoide* y es un modo de oscilación que también aparece en aeronaves de ala fija.

Si lo evaluamos en la primera oscilación de la trayectoria correspondiente a  $E = 1$  de la Figura 3 tenemos la siguiente expresión

$$T = \frac{2\pi}{\sqrt{\frac{1}{7,524 \cdot 10^3} (1 - 0,875^2)}} = 964 \quad [\text{s}] = 16 \quad [\text{min}] . \quad (14)$$

aproximadamente lo mismo que dura la primera oscilación.

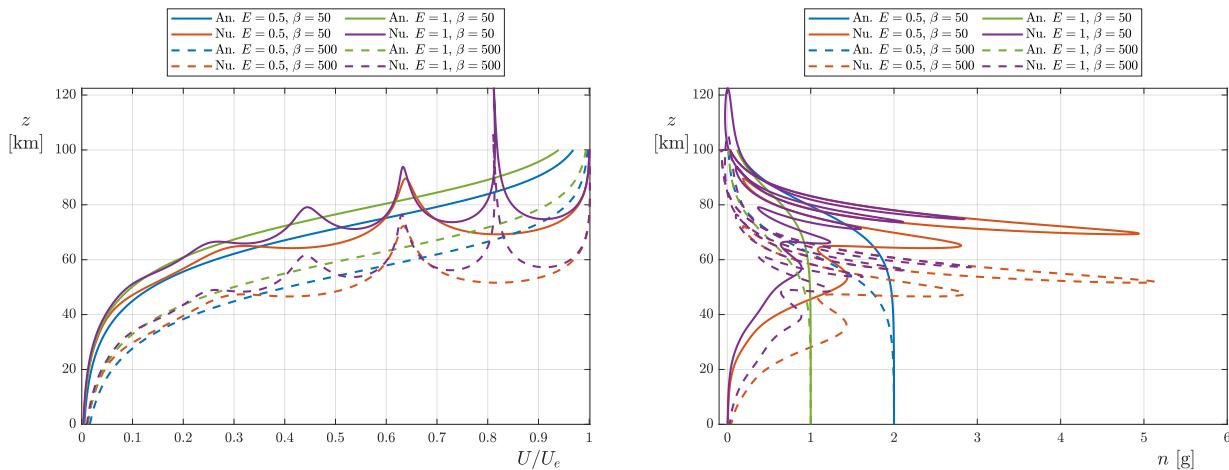


Figura 6: Comparación de resultados en la entrada sustentada entre el método numérico y analítico.

La expresión obtenida anteriormente muestra la oscilación en la altura, pero también afecta a todas las variables, haciendo que el resultado numérico se aleje notablemente de la solución analítica para el caso de reentrada con sustentación. Como este fenómeno solo aparece cuando hay sustentación, en la entrada balística no hay oscilaciones, tal y como era de esperar.

Para concluir con el trabajo se va a comentar la Figura 7. En esta se puede resumir todo lo comentado anteriormente.

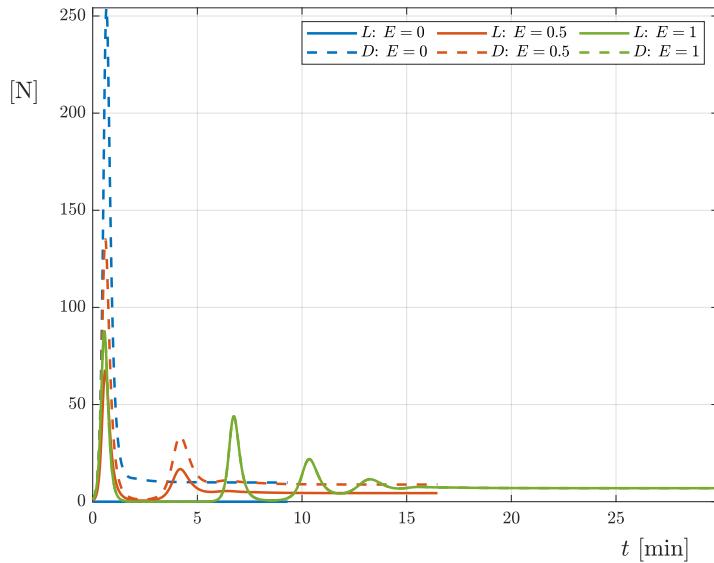


Figura 7: Sustentación y resistencia aerodinámica a lo largo de la reentrada para la segunda combinación de condiciones iniciales y  $\beta = 50$

En esta figura se ha representado la evolución de la sustentación y la resistencia aerodinámica durante la reentrada para la tercera combinación de condiciones iniciales. Como se ha visto, para entrada balística no se genera sustentación y el valor máximo de la resistencia es el mayor de todos, llegando en la fase final a un estacionario.

Por otro lado, en las curvas de reentrada sustentada, se puede ver el cambio brusco en la sustentación que provoca el *fugoide* y este pico en la sustentación coincide con el instante en el que la cápsula comienza a ascender en la Figura 3a.

Finalmente, cuando desaparece el fugoide la cápsula llega a un estacionario hasta que aterriza.

## A. Código

### A.1. Programa principal

```
1 clc
2 clear
3 close all
4
5
6 %% GLOBAL VARIABLES
7
8 global g0; g0 = 9.81; % [m/s^2]
9 global Rt; Rt = 6378e3; % [m]
10
11
12 %% PARAMETERS & IC
13
14 % Capsule parameters
15 E = [0 0.5 1];
16 beta = [50 250 500];
17 % Intial condition
18 y0_set = [[100e3, 7.5e3, deg2rad(-5)],...
19 [100e3, 8.5e3, deg2rad(-5)],...
20 [100e3, 7.5e3, deg2rad(-10)']];
21
22 % RK4 time step
23 Dt = 0.1; % [s]
24
25
26 %% PLOT DEFINITION
27
28 colors = [0, 0.4470, 0.7410;
29 0.8500, 0.3250, 0.0980;
30 0.4660, 0.6740, 0.1880;
31 0.4940, 0.1840, 0.5560];
32
33 line_style = {'-', '-.', ':'};
34 leg_str = ["'$E = ' num2str(E(e)) ' $ ; $\\beta = ' num2str(beta(b)) '$'"];
35 PDF_str = ["['Figures/Num_Fig_' num2str(f) '_ue_', num2str(y0(2)/100), '_gammae_', ...
36 num2str(rad2deg(y0(3))) ]"];
37
38 for y = 1:size(y0_set,2)
39
40 y0 = y0_set(:,y);
```

```
41
42 % Numeric solution
43 numeric = Function_Numeric(beta, E, y0, Dt);
44 numerics{y} = numeric;
45
46 % Analytic solution
47 [ballistic, glide] = Function_Analytic(beta, E, y0);
48 analytics(y).ballistic = ballistic;
49 analytics(y).glide = glide;
50
51
52 % Numeric
53 for b = 1:length(beta)
54     for e = 1:length(E)
55
56         h{1} = figure(1);
57         hold on
58         plot(numeric(b,e).t/60, rad2deg(numeric(b,e).gamma),...
59               'line_style{b}', 'Color', colors(e,:), 'LineWidth', 1.5,...
60               'DisplayName', eval(leg_str))
61         axis([0, max([numeric(:,:,tx])/60, -90, ...
62               rad2deg(max([numeric(:,:,gammax)]))])
63         grid on; box on;
64         xlabel('$t$ [min]', 'Interpreter', 'Latex')
65         ylabel({'$\gamma$'; '$\mathbf{\omega}$'}, 'Interpreter', 'Latex')
66
67         h{2} = figure(2);
68         hold on
69         plot(numeric(b,e).t/60, ...
70               numeric(b,e).u/numeric(b,e).u0, ...
71               'line_style{b}', 'Color', colors(e,:), 'LineWidth', 1.5,...
72               'DisplayName', eval(leg_str))
73         axis([0, max([numeric(:,:,tx))/60, 0, ...
74               max([numeric(:,:,ux))/numeric(b,e).u0])
75         grid on; box on;
76         xlabel('$t$ [min]', 'Interpreter', 'Latex')
77         ylabel({'$U$'}, 'Interpreter', 'Latex')
78
79         h{3} = figure(3);
80         hold on
81         plot(numeric(b,e).t/60, numeric(b,e).z/1000, ...
82               'line_style{b}', 'Color', colors(e,:), 'LineWidth', 1.5,...
83               'DisplayName', eval(leg_str))
84         axis([0, max([numeric(:,:,tx))/60, 0, max([numeric(:,:,zx))/1000])
85         grid on; box on;
86         xlabel('$t$ [min]', 'Interpreter', 'Latex')
87         ylabel({'$z$'; '[km]'}, 'Interpreter', 'Latex')
```

```

86
87     h{4} = figure(4);
88         hold on
89         plot(numeric(b,e).t/60,-numeric(b,e).a/9.81,line_style{b},...
90             'Color', colors(e,:), 'LineWidth', 1.5,...
91             'DisplayName', eval(leg_str))
92         axis([0, max([numeric(:, :).tx])/60,-.1, -min([numeric(:, :).ax])/9.81])
93         grid on; box on;
94         xlabel('$t$ [min]', 'Interpreter', 'Latex')
95         ylabel({'$n$'; '[g]'}, 'Interpreter', 'Latex')

96
97     end
98 end
99
100 for f = 1:4
101     Save_as_PDF(h{f}, eval(PDF_str), 'vertical')
102     figure(f);
103         legend('Location', 'NorthOutside', 'Interpreter', 'Latex', ...
104             'NumColumns', 3)
105     h{f} = figure(f);
106     h{f}.Position = [7.0729, 5.8021, 1.15*5.8333, 1.15*4.3750];
107     Save_as_PDF(h{f}, eval(PDF_str), 'horizontal')
108 end
109
110 close all
111 clear h
112
113
114 %% ANALYTIC AND NUMERIC COMPARISON
115
116
117 close all
118 line_style = {'-', ':', '--'};
119
120 % BALLISTIC
121 e = 1;
122 leg_str_ba = ["[An. $\gamma_e =' num2str(rad2deg(y0_set(3,y))) '$, $\beta =' ...
123     num2str(beta(b)) '$' ]";
124 leg_str_bn = ["[Nu. $\gamma_e =' num2str(rad2deg(y0_set(3,y))) '$, $\beta =' ...
125     num2str(beta(b)) '$' ]";
126
127 % Speed
128 h = figure(1);
129     hold on
130     for y = 1:2:3
131         for b = 1:2:3

```

```
130 plot(analytics(y).ballistic(b).u, analytics(y).ballistic(b).z/1000, ...
131 line_style{b}, 'Color', colors(y,:), 'LineWidth', 1.5, ...
132 'DisplayName', eval(leg_str_ba));
133 plot(numerics{y}(b,e).u/numerics{y}(b,e).u0, ...
134 numerics{y}(b,e).z/1000, line_style{b}, ...
135 'Color', colors(y+1,:), 'LineWidth', 1.5, ...
136 'DisplayName', eval(leg_str_bn))
137 end
138 end
139 axis([0, max([numerics{y}(:,:,ux)])/numerics{y}(b,e).u0, 0, ...
140 max([numerics{y}(:,:,zx)])/1000])
141 grid on; box on;
142 legend('Location', 'NorthOutside', 'Interpreter', 'Latex', 'NumColumns', 2)
143 xlabel('$U/U_e$', 'Interpreter', 'Latex')
144 ylabel({'$z$'; '[km]'}, 'Interpreter', 'Latex')

145 Save_as_PDF(h, 'Figures/Comparison_Ballistic_U', 'vertical')
146 figure(1);
147 legend('Location', 'NorthOutside', 'Interpreter', 'Latex', 'NumColumns', 2)
148 h = figure(1);
149 h.Position = [7.0729, 5.8021, 1.15*5.8333, 1.15*4.3750];
150 Save_as_PDF(h, 'Figures/Comparison_Ballistic_U', 'horizontal')

151
152
153 % Accleration
154 h = figure(2);
155 hold on
156 for y = 1:2:3
157     for b = 1:2:3
158         plot(analytics(y).ballistic(b).n, analytics(y).ballistic(b).z/1000, ...
159 line_style{b}, 'Color', colors(y,:), 'LineWidth', 1.5, ...
160 'DisplayName', eval(leg_str_ba));
161         plot(-numerics{y}(b,e).a/9.81, numerics{y}(b,e).z/1000, ...
162 line_style{b}, 'Color', colors(y+1,:), 'LineWidth', 1.5, ...
163 'DisplayName', eval(leg_str_bn))
164     end
165 end
166 axis([-0.1, -min([numeric(:,:,ax))/9.81, 0, max([numerics{y}(:,:,zx)])/1000])
167 grid on; box on;
168 legend('Location', 'NorthOutside', 'Interpreter', 'Latex', 'NumColumns', 2)
169 xlabel('$n$ [g]', 'Interpreter', 'Latex')
170 ylabel({'$z$'; '[km]'}, 'Interpreter', 'Latex')

171
172 Save_as_PDF(h, 'Figures/Comparison_Ballistic_a', 'vertical', 0, 10)
173 figure(2);
174     legend('Location', 'NorthOutside', 'Interpreter', 'Latex', 'NumColumns', 2)
175 h = figure(2);
```

```

176     h.Position = [7.0729, 5.8021, 1.15*5.8333, 1.15*4.3750];
177     Save_as_PDF(h, 'Figures/Comparison_Ballistic_a', 'horizontal')
178
179
180 % GLIDE
181
182 close all
183
184 leg_str_ga = "[ 'An. $E =' num2str(E(e)) '$, $\beta =' num2str(beta(b)) '$ ]";
185 leg_str_gn = "[ 'Nu. $E =' num2str(E(e)) '$, $\beta =' num2str(beta(b)) '$ ]";
186
187 % Speed
188 h = figure(1);
189 hold on
190 y = 1;
191 for e = 2:3
192     for b = 1:2:3
193         plot(analytics(y).glide(b,e).u, analytics(y).glide(b,e).z/1000, ...
194             line_style{b}, 'Color', colors(2*e-3,:), 'LineWidth', 1.5, ...
195             'DisplayName', eval(leg_str_ga));
196         plot(numerics{y}(b,e).u/numerics{y}(b,e).u0, ...
197             numerics{y}(b,e).z/1000, line_style{b}, ...
198             'Color', colors(2*e-2,:), 'LineWidth', 1.5, ...
199             'DisplayName', eval(leg_str_gn))
200     end
201 end
202 axis([0, max([numerics{y}(:,:,ux)])/numerics{y}(b,e).u0, 0, ...
203       max([numerics{y}(:,:,zx)])/1000])
204 grid on; box on;
205 legend('Location', 'NorthOutside', 'Interpreter', 'Latex', 'NumColumns', 2)
206 xlabel('$U/U_e$', 'Interpreter', 'Latex')
207 ylabel({'$z$'; '[km]'}, 'Interpreter', 'Latex')

208 Save_as_PDF(h, 'Figures/Comparison_Glide_U', 'vertical')
209 figure(1);
210 legend('Location', 'NorthOutside', 'Interpreter', 'Latex', 'NumColumns', 2)
211 h = figure(1);
212 h.Position = [7.0729, 5.8021, 1.15*5.8333, 1.15*4.3750];
213 Save_as_PDF(h, 'Figures/Comparison_Glide_U', 'horizontal')

214 % Acceleration
215
216 h = figure(2);
217 hold on
218 for e = 2:3
219     for b = 1:2:3
220         plot(analytics(y).glide(b,e).n, analytics(y).glide(b,e).z/1000, ...
221             line_style{b}, 'Color', colors(2*e-3,:), 'LineWidth', 1.5, ...

```

```

222         'DisplayName', eval(leg_str_ga));
223 plot(-numerics{y}(b,e).a/9.81,numerics{y}(b,e).z/1000, ...
224     line_style{b}, 'Color', colors(2*e-2,:), 'LineWidth', 1.5, ...
225     'DisplayName', eval(leg_str_gn))
226 end
227 end
228 axis([-0.1, 6, 0, max([numerics{y}(:, :).zx])/1000])
229 grid on; box on;
230 legend('Location', 'NorthOutside', 'Interpreter', 'Latex', 'NumColumns', 2)
231 xlabel('$n$ [g]', 'Interpreter', 'Latex')
232 ylabel({'$z$'; '[km]'}, 'Interpreter', 'Latex')

233
234 Save_as_PDF(h, 'Figures/Comparison_Glide_a', 'vertical', 0, 10)
235 figure(2);
236 legend('Location', 'NorthOutside', 'Interpreter', 'Latex', 'NumColumns', 2)
237 h = figure(2);
238 h.Position = [7.0729, 5.8021, 1.15*5.8333, 1.15*4.3750];
239 Save_as_PDF(h, 'Figures/Comparison_Glide_a', 'horizontal')

240
241
242 %% LIFT
243
244 L = {};
245 D = {};
246 h = figure();
247 hold on
248 for e = 1:length(E)
249     [rho, L, D] = Function_rho(numerics{3}(1,e).z);
250     L = cat(1, L, rho.*numerics{3}(1,e).u.^2*E(e)/(2*beta(1)));
251     D = cat(1, D, rho.*numerics{3}(1,e).u.^2/(2*beta(1)));
252     plot(numerics{3}(1,e).t/60, L{e}(:, ), 'Color', colors(e, :), ...
253           'LineWidth', 1.5, 'DisplayName', ['$L$', '$E' = num2str(E(e))]);
254     plot(numerics{3}(1,e).t/60, D{e}(:, ), line_style{3}, 'Color', colors(e, :), ...
255           'LineWidth', 1.5, 'DisplayName', ['$D$', '$E' = num2str(E(e))]);
256 end
257 axis([0, max([numerics(:, :).tx])/60, 0, max(D{1}(:, ))])
258 Save_as_PDF(h, 'Figures/Lift', 'horizontal')
259 h.Position = [7.0729, 5.8021, 1.15*5.8333, 1.15*4.3750];
260 grid on; box on;
261 xlabel('$t$ [min]', 'Interpreter', 'Latex')
262 ylabel({'[N]'}, 'Interpreter', 'Latex')
263 legend('Interpreter', 'Latex', 'Location', 'NorthEast', 'NumColumns', 3)
264 Save_as_PDF(h, 'Figures/Lift', 'horizontal')

265
266 %% FUGOIDE
267
268 T = 2*pi/( sqrt( (1 - .825^2)/((7.524e3) ) ) ) / 60

```



POLITÉCNICA

UNIVERSIDAD  
POLITÉCNICA  
DE MADRID

AAV&FR: Trabajo 3

## Referencias

- [1] S. Franchini, Dinámica de la reentrada (2020-21).  
URL [www.idr.upm.es](http://www.idr.upm.es)
- [2] Fourth Order Runge-Kutta.  
URL <https://lpsa.swarthmore.edu/NumInt/NumIntFourth.html>