

---

# PROPULSIÓ

## ESTUDI PARAMÈTRIC DE L'ASCENS VERTICAL D'UN COET

---

Alumnes: CRISTIAN ASENSIO GARCÍA, YI QIANG JI ZHANG,  
PEDRO LÓPEZ SANCHÀ, SANTIAGO VILLARROYA CALAVIA,

Professor: JOSEP ORIOL LIZANDRA DALMASES

Enginyeria en Tecnologies Aeroespacials  
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa  
Universitat Politècnica de Catalunya

4 de febrer de 2021

## Índex

<b>1</b>	<b>Introducció</b>	<b>2</b>
<b>2</b>	<b>Plantejament</b>	<b>3</b>
<b>3</b>	<b>Hipòtesis inicials</b>	<b>4</b>
<b>4</b>	<b>Resolució</b>	<b>5</b>
4.1	Càlcul de la velocitat característica $c^*$ i flux màssic per unitat d'àrea $\frac{\dot{m}}{A_t}$ . . . . .	5
4.2	Càlcul del coeficient d'empenta $C_F$ . . . . .	5
<b>5</b>	<b>Representacions gràfiques</b>	<b>8</b>
5.1	Resultats negligint la resistència aerodinàmica . . . . .	8
5.2	Resultats amb resistència aerodinàmica . . . . .	10
<b>6</b>	<b>Relació d'àrees òptima</b>	<b>16</b>
<b>7</b>	<b>Evolució de les forces durant la trajectòria</b>	<b>18</b>
<b>8</b>	<b>Alçada i velocitats finals front a la relació d'àrees</b>	<b>20</b>
<b>A</b>	<b>Coefficient de resistència aerodinàmica</b>	<b>21</b>
<b>B</b>	<b>Descripció del Criteri de Summerfield</b>	<b>22</b>
<b>C</b>	<b>Integració numèrica Runge-Kutta RK4</b>	<b>23</b>
C.1	Resolució numèrica de sistemes d'equacions diferencials . . . . .	23
C.2	Resolució de la trajectòria sense resistència aerodinàmica . . . . .	24
C.3	Resolució de la trajectòria amb resistència aerodinàmica . . . . .	25
<b>D</b>	<b>Codi del projecte</b>	<b>26</b>
D.1	Funció pel càlcul de les temperatures de cada capa atmosfèrica . . . . .	27
D.2	Funció pel càlcul de les condicions atmosfèriques . . . . .	28
D.3	Funció pel càlcul del $MFP$ . . . . .	30
D.4	Funció per representar el $MFP$ . . . . .	30
D.5	Programa per evolució <b>sense</b> $Drag$ . . . . .	32
D.6	Programa per evolució <b>amb</b> $Drag$ . . . . .	37
D.7	Programa per evolució de forces . . . . .	42
D.8	Programa per càlcul de l'alçada i velocitat final . . . . .	48

## 1 Introducció

Al llarg dels últims anys ha aparegut un creixent interès en realitzar viatges espacials. Això ha requerit cada vegada més i millors mètodes per la generació de forces externes propulsores capaces de generar empentes que permetin escapar del camp gravitatori terrestre o posar un vehicle en òrbita. Amb aquest objectiu van néixer els motors coet, un tipus de motor capaç de generar empentes considerables i que ha evolucionat gràcies a una continua recerca en aquest sector. Degut a que el funcionament de la majoria d'aquests motors es basa en l'expulsió de combustible i comburent a grans velocitats i temperatures, no es tracta d'un sistema molt eficient. Per aquesta raó en els últims anys s'ha investigat en alternatives per millorar el seu rendiment. Algunes d'aquestes alternatives consisteixen en la incorporació de múltiples etapes, coets híbrids i inclús coets reutilitzables.

En aquest treball que es presenta a continuació, s'estudien l'influència de les propietats geomètriques i constructives en el vol de la primera etapa d'un motor coet, és a dir, com es veuen influenciades les velocitats i altures finals. Així mateix, es pretén presentar una idea general sobre el funcionament d'aquest motors i de com es poden optimitzar els seus resultats.

## 2 Plantejament

Es considera un vehicle llançador amb una massa inicial de 540 t, amb dues etapes, la primera equipada amb un coet amb les següents característiques:

- Pressió del combustor:  $p_c = 100 \text{ atm}$
- Temperatura del combustor:  $T_c = 3500 \text{ K}$
- Propietats del gas propulsant:  $\gamma = 1.25$ ;  $MW = 16 \text{ g} \cdot \text{mol}^{-1}$
- Flux màssic de propulsant:  $\dot{m} = 2000 \text{ kg} \cdot \text{s}^{-1}$
- Temps de combustió:  $t_b = 60 \text{ s}$ .

L'àrea transversal de referència per la resistència aerodinàmica és  $S_{\text{ref}} = 25 \text{ m}^2$  i el coeficient de resistència aerodinàmica  $C_D$  es funció del número de Mach, com s'indica a l'annex A. Per un ascens vertical assumint atmosfera ISA, es demana el següent:

1. Representar l'altitud i la velocitat en funció del temps, per relacions d'àrees  $A_e/A_t = 20, 40, 60, 80$ , ignorant la resistència aerodinàmica.
2. Repetir els càlculs i les representacions gràfiques considerant la resistència aerodinàmica.
3. A partir dels resultats de l'apartat anterior, indicar quina de les relacions d'àrees és la millor elecció (ignorant els efectes del pes).

### 3 Hipòtesis inicials

Per tal de simplificar el problema plantejat, es fan una sèrie d'hipòtesis que es presenten a continuació:

- **Model quasi-dimensional:** Les magnituds varien de forma suau al llarg de la secció perpendicular a l'eix de la tovera. En altres paraules, les propietats i magnituds varien de forma quasiestàtica.
- **Adiabàtic i isentròpic:** El flux es considera adiabàtic i isentròpic si es troben allunyats de la pared, ja que estan fora de la capa límit.
- **Gas ideal i caloríficament perfecte** i amb propietats del gas conegudes: Massa molecular, coeficients de transferència de calor i la constant adiabàtica.

$$R = \frac{R_u}{MW} \quad \gamma = \frac{c_p}{c_v}$$

on  $R_u = 8.314472 \text{ Jmol}^{-1} \cdot \text{K}^{-1}$ ,  $MW$ : pes molecular i la constant adiabàtica  $\gamma$  és la relació entre la calor específica a pressió constant  $c_p$  i la calor específica a volum constant  $c_v$ .

- **Velocitats negligibles** a la cambra de combustió. Això significa, que a efectes pràctics les propietats d'estancament i les propietats absolutes són virtualment iguals a la cambra.
- **Tovera ofegada:** Es considera que la tovera està *choked* ja que la pressió de la cambra de combustió  $p_c$  és en tot moment molt major a la pressió ambient.

## 4 Resolució

L'objectiu de l'anàlisi és trobar la velocitat respecte el temps i l'altura del coet en cada moment. Com s'ha comentat anteriorment, s'estudiarà dos casos. En la primera, es considerarà nuls els efectes de resistència aerodinàmica  $D$  i considerant els efectes Drag. El procediment per ambdós mètodes és anàleg. Posteriorment, es compararan els resultats d'ambdós casos i s'observarà si realment aquest efecte implica unes pèrdues considerables o es pot negligir sense suposar una aproximació poc precisa [vegeu apartat 7].

### 4.1 Càlcul de la velocitat característica $c^*$ i flux màssic per unitat d'àrea $\frac{\dot{m}}{A_t}$

Primerament, és necessari trobar la constant del gas  $R$ :

$$R = \frac{R_u}{MW} = \frac{8.31432}{0.016} = 512.6450 \text{ J/kg}^{-1} \text{ mol}^{-1} \quad (4.1)$$

on  $R_u = 8.31432 \text{ J/mol}^{-1} \text{ K}^{-1}$  és la constant universal dels gasos ideals i massa molecular del gas en concret  $MW = 0.016$ . Seguidament, un cop s'ha calculat la constant del gas  $R$ , la **velocitat característica**  $c^*$  en aquest es pot obtenir com:

$$c^* = \frac{\sqrt{RT_c}}{\bar{m}(M_t = 1, \gamma)} = \frac{\sqrt{512 \cdot 3500}}{0.6581} = 2049.2527 \text{ m s}^{-1} \quad (4.2)$$

A continuació, es pot obtenir el **flux màssic per unitat d'àrea** de la gola  $\frac{\dot{m}}{A_t}$  com la relació entre la pressió de cambra  $p_c$  i la velocitat característica  $c^*$ :

$$\frac{\dot{m}}{A_t} = \frac{p_c}{c^*} = \frac{100 \cdot 101325}{2049.2527} = 4944.4854 \text{ kg s}^{-1} \text{ m}^{-1} \quad (4.3)$$

### 4.2 Càlcul del coeficient d'empenta $C_F$

En segon lloc, és necessari calcular el coeficient d'empenta  $C_F$  en el moment de l'enlairament. És important comprovar però, si el flux està després en aquesta condició. Per aquest motiu, s'estudiarà en primer lloc si la pressió de la sortida és superior o inferior a 0.4 vegades la pressió atmosfèrica.

La relació d'àrees d'àrees de la sortida de la tovera  $A_e$  i de la gola  $A_t$  són conegudes. Es pot demostrar que aquesta relació d'àrees és també la relació del *Mass Flow Parameter*  $\bar{m}$  de la sortida i de la gola. És a dir,

$$\frac{A_e}{A_t} = \frac{\bar{m}(M_t = 1, \gamma)}{\bar{m}(M_e, \gamma)} \quad (4.4)$$

Com el terme  $\bar{m}(M_t = 1, \gamma)$  és conegut es pot aïllar directament el valor de  $\bar{m}(M_e, \gamma)$ . A tall d'exemple, es farà els càlculs amb la primera relació d'àrees  $\frac{A_e}{A_t}$

$$\frac{A_e}{A_t} = 20 \quad \Rightarrow \quad \bar{m}(M_e, \gamma) = \frac{\bar{m}(M_t = 1, \gamma)}{\frac{A_e}{A_t}} = \frac{0.6581}{20} = 0.0329 \quad (4.5)$$

Fent ús de l'expressió del *Mass Flow Parameter* o de les Taules [3] es pot obrtenir el valor de Mach associat. En particular, pel que fa referència al nostre estudi, s'ha optat per usar una resolució numèrica de (4.6) per tal d'obtenir una major precisió. <sup>1</sup>

$$\bar{m}(M, \gamma) = \sqrt{\gamma} \frac{M}{\left(a + \frac{\gamma-1}{2} M^2\right)^{\frac{\gamma+1}{2(\gamma-1)}}} \quad (4.6)$$

---

<sup>1</sup>Cal destacar que la funció del *MPF* té dues solucions per un mateix Mach. La solució del cas analitzat amb significat físic és aquella en la qual el Mach de sortida és supersònic  $1 \leq M_e \leq 6$ .

La resolució numèrica mitjançant MATLAB dóna com a solució el Mach de sortida. De la mateixa manera, a partir de les taules [3], el valor exacte de 0.0329 no és present i s'hauria d'interpol·lar per obtenir un resultat més exacte. Tanmateix, el valor de Mach de sortida es troba dins del rang de valors de la Taula amb la qual cosa verifica el resultat obtingut.

$$M_e = 3.9746 \quad (4.7)$$

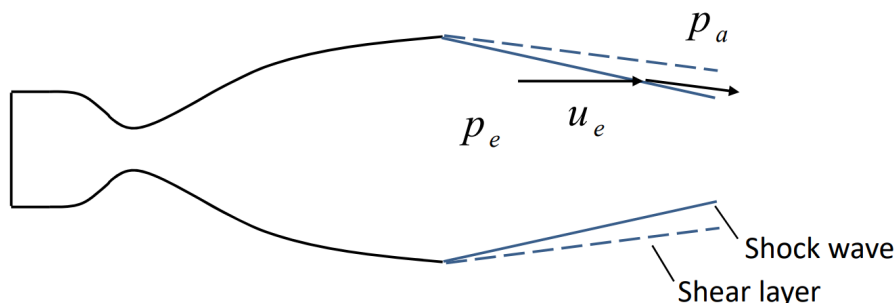
Un cop obtingut el Mach de sortida  $M_e$ , fent ús de la relació isentròpica de la pressió entre la pressió de cambra  $p_c$  i la pressió de sortida es pot calcular aquesta última ja que només depèn del número de Mach i de la relació de calors del fluid  $\gamma$ .

$$p_e = \frac{p_c}{\left(1 + \frac{\gamma + 1}{2} M_e^2\right)^{\frac{\gamma}{\gamma - 1}}} = 4.3502 \cdot 10^4 \text{ Pa} = 0.4293 \text{ atm} \quad (4.8)$$

Un cop tenim la pressió de sortida, cal comprovar la condició del flux de sortida i observar si es tracta de:

- Tovera Sub-expansionada
- Tovera Adaptada
- Tovera Sobre-expansionada

En efecte, en cas de tenir una tovera sobre-expansionada pot donar-se el cas d'una aparició d'ones de xoc suficientment potents com per produir un despreniment de la capa límit a les parets internes de la tovera. D'aquest fenomen se'n diu que la tovera es troba **Altament sobre-expansionada**.



**Figura 4.1** Esquema del flux en una tovera sobre-expansionada. A partir de [3]

Per tant, cal aplicar el **criteri de Summerfield B** la qual indica que per toveres altament sobre-expansionades, el flux en la tovera es desprendreà en el moment en què la pressió sigui aproximadament 0.4 vegades la pressió de ambient. Aigües avall d'aquest punt, la pressió interna de la tovera serà pràcticament la mateixa que la pressió atmosfèrica, amb la qual cosa la diferencia de pressions d'ambdós costats de la tovera es cancel·len. Aquest despreniment del flux produeix que la succió aigües avall del es redueixi millorant significativament l'empenta.

Mitjançant un procés iteratiu, en el que es calcula la pressió de sortida per a cada instant de temps  $0 \leq t \leq t_{final}$  i per totes les relacions d'àrees  $\frac{A_e}{A_t}$ .

En funció de la pressió de sortida teòrica  $p_e$  i la pressió ambient en aquell instant  $p_a$ , es podrà determinar a pressió de sortida real  $p'_e$ . Per consegüent, la  $p'_e$  es determinarà segons:

- a)  $p_e < 0.4p_a$

Segons el criteri de Summerfield, l'àrea de sortida inicial es substituirà per una àrea de sortida

*efectiva*. Per tant,

$$p_e = 0.4p_a = \frac{p_c}{\left(1 + \frac{\gamma+1}{2}M_e'^2\right)^{\frac{\gamma}{\gamma-1}}} \implies M_e' = \sqrt{\frac{2}{\gamma-1} \left[ \left(\frac{p_c}{0.4p_a}\right)^{\frac{\gamma-1}{\gamma}} - 1 \right]} \quad (4.9)$$

Amb aquest nou Mach de sortida  $M_e'$ , es calcularà la relació d'àrees efectiva:

$$\frac{A_e'}{A_t} = \frac{\bar{m}(M_t = 1, \gamma)}{\bar{m}(M_e', \gamma)} \quad (4.10)$$

Amb aquesta nova relació d'àrees, es pot obtenir el coeficient d'empenyiment  $C_F$  amb l'expressió següent 4.11:

$$C_F = \underbrace{\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} \frac{\gamma M_e' + 1/M_e'}{\sqrt{1 + \frac{\gamma-1}{2}(M_e')^2}}}_{C_{Fvac}} - \frac{p_a}{p_c} \frac{\bar{m}(M_t = 1, \gamma)}{\bar{m}(M_e', \gamma)} \quad (4.11)$$

De forma que avaluant  $C_F$  a  $M_e'$  s'obté el coeficient d'empenyiment del coet.

b)  $p_e \geq 0.4p_a$

Per altra banda, en aquesta condició, la tovera es troba o adaptada o sub-expansionada. El procés és anàleg a la condició anterior i la diferència rau en que no caldrà considerar una àrea de sortida efectiva.

$$\frac{A_e}{A_t} \implies \bar{m}(M_e, \gamma) = \frac{\bar{m}(M_t = 1, \gamma)}{\frac{A_e}{A_t}} \quad (4.12)$$

Ara,

$$C_F = \underbrace{\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} \frac{\gamma M_e' + 1/M_e'}{\sqrt{1 + \frac{\gamma-1}{2}(M_e')^2}}}_{C_{Fvac}} - \frac{p_a}{p_c} \frac{\bar{m}(M_t = 1, \gamma)}{\bar{m}(M_e', \gamma)} \quad (4.13)$$

Amb aquestes dades calculades per a cada alçada i amb les distribucions de pressió i temperatura de a cada alçada. Aplicant el mètode d'integració numèrica per Runge-Kutta que s'explicarà posteriorment en l'apartat C

Les dues equacions a integrar són:

$$\frac{dh}{dt} = V \quad (4.14)$$

$$\frac{dV}{dt} = \frac{F - D}{m(t)} - g = a \quad (4.15)$$

I les condicions inicials són:

$$h(0) = 0$$

$$V(0) = 0$$



## 5 Representacions gràfiques

Una vegada calculats els paràmetres i les característiques principals del coet, en funció de les diferents relacions d'àrees de sortida, es procedeix a realitzar un estudi de l'evolució de la trajectòria del coet durant el seu enlairament.

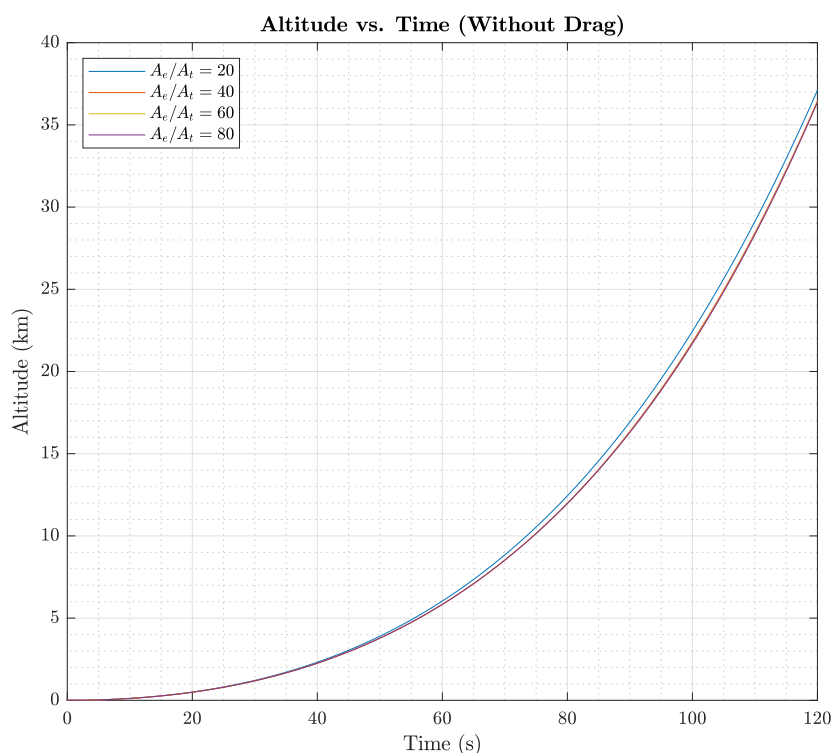
Com s'ha comentat anteriorment, el principal objectiu d'aquest estudi es ser capaçs d'avaluar amb prou exactitud l'alçada i la velocitat final a la que arribarà el coet, per a un temps d'encesa del motor concret, tot tenint en compte l'efecte de la relació d'àrees de sortida i de la gola.

De manera simultània es pretén observar l'efecte de la resistència aerodinàmica sobre el valor final de les variables alçada i velocitat.

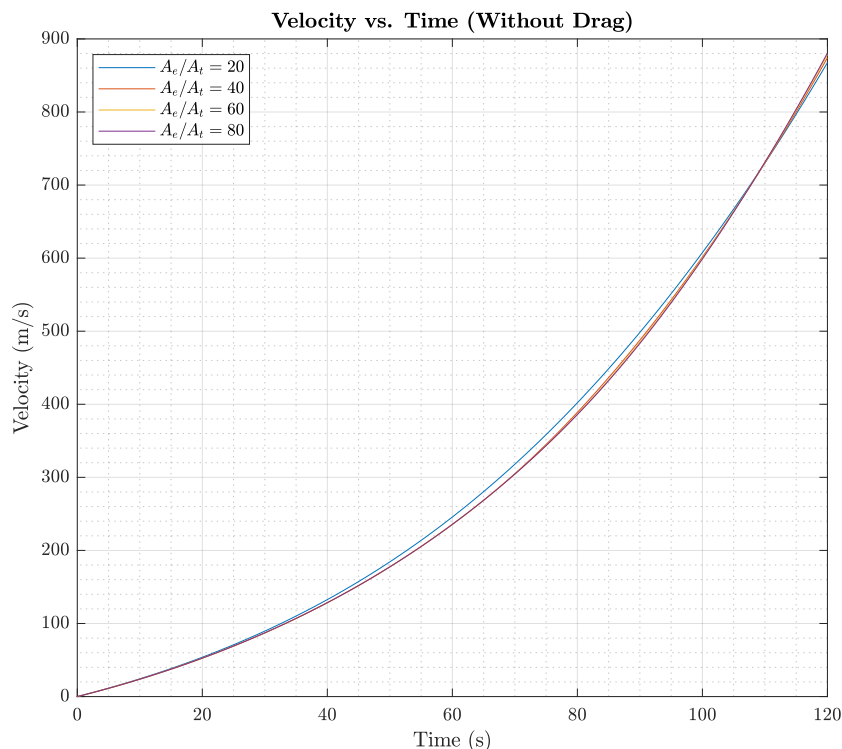
Per això mateix es resolen mitjançant un esquema d'integració numèrica com el que s'explica a l'annex C.1, els dos sistemes d'equacions diferencials que resulten de treballar les equacions dinàmiques i cinemàtiques del coet, sense drag i amb drag, i que es mostren als annexos C.2 i C.3 respectivament.

Abans de mostrar les diferents representacions gràfiques obtingudes per a cada cas, es destaca el fet que tot i què a l'enunciat del treball que s'ens va facilitar, només es demanaven els resultats de la alçada i la velocitat al cap d'un temps d'encesa de 60 segons, amb la intenció d'obtenir conclusions de caràcter molt més genèric, es proposa l'estudi general per a un temps d'encesa equivalent a 2 minuts. No obstant, al final d'aquest apartat també es presenten els valors finals i l'evolució de les variables alçada i velocitat que el coet seria capaç d'assolir al cap de 1 minut, 2 minuts i 2 minuts 40 segons.

### 5.1 Resultats negligint la resistència aerodinàmica



**Figura 5.1** Evolució de la variable alçada durant la trajectòria d'enlairament menyspreant el Drag



**Figura 5.2** Evolució de la variable velocitat durant la trajectòria d'enlairament menyspreant el Drag

Tal com es mostra a les dues gràfiques anteriors les evolucions de l'alçada i la velocitat al llarg del temps, durant el procés d'enlairament del coet, presenten tendències creixents realment similars.

Un dels fet que resulta més destacable d'aquests resultats es que per al cas de les relacions d'àrees de  $A_e/A_t = 40, 60$  i  $80$  tant la alçada com la velocitat assoleixen valors gairebé idèntics durant tota la trajectòria. Trobem la seva justificació en el fet de què degut a la gran superfície de sortida que presenten aquestes toveres respecte a la superfície de la gola, aquestes es troben pràcticament tota l'estona en condicions de tovera sobreexpansionada. Com s'ha explicat a l'apartat anterior si seguim el *Criteri de Summerfield*, quan la tovera està massa sobreexpansionada, com es dona en aquests tres casos, es produirà un desprendiment de flux en la secció de la tovera per la que es tingui una pressió equivalent a 0.4 vegades la pressió exterior. Fixant-nos en les expressions a) això provocarà que tant la relació d'àrees efectiva com el Mach efectiu de sortida de la tovera es vegin disminuïts, de la mateixa manera per a les tres relacions de  $A_e/A_t = 40, 60$  i  $80$  i per aquest motiu els seus valors de alçades i velocitats en funció del temps siguin tan propers durant tot el interval de temps en el que s'estudien.

En segon lloc, et pot apreciar tan a la gràfica de la velocitat com sobretot a la gràfica de l'alçada com la tendència creixent que presenten té un pendent cada vegada més pronunciat a mesura que avança el temps. Això té lloc gracies a dos aspectes concrets:

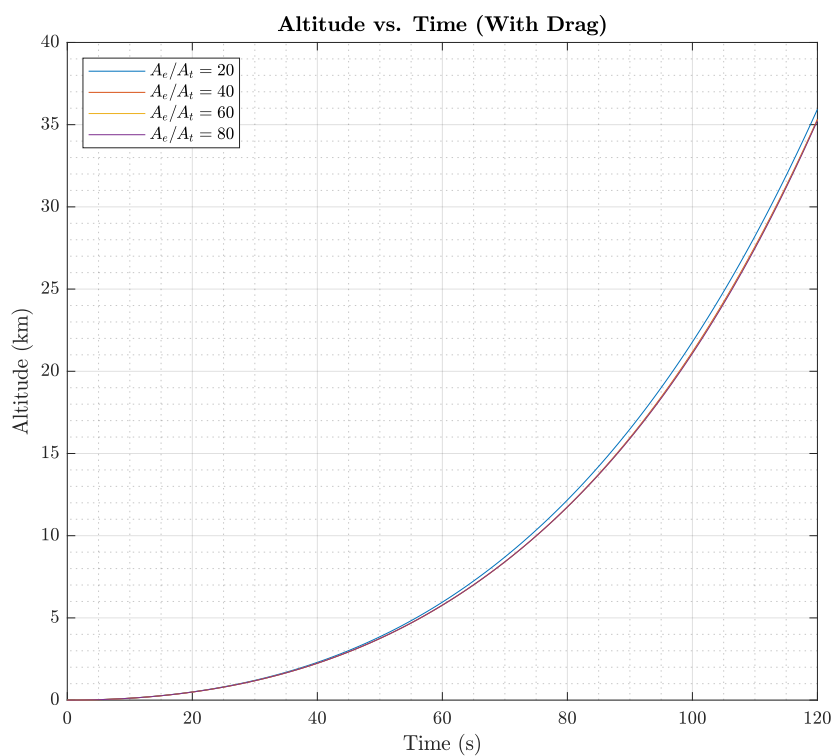
- L'efecte de la força de empenyiment vas augmentant a mesura que augmenta l'alçada i en conseqüència es redueix la pressió atmosfèrica exterior.
- La contribució negativa del pes cada vegada es més petita com a causa de l'ejecció del combustible necessari per proporcionar l'empenyiment.

Finalment s presenten en la següent taula els valors finals de l'alçada, la velocitat i el número de Mach que el coet objecte d'estudi amb les 4 diferents relacions d'àrees seria capaç d'assolir, si es menysprea la contribució de la resistència aerodinàmica.

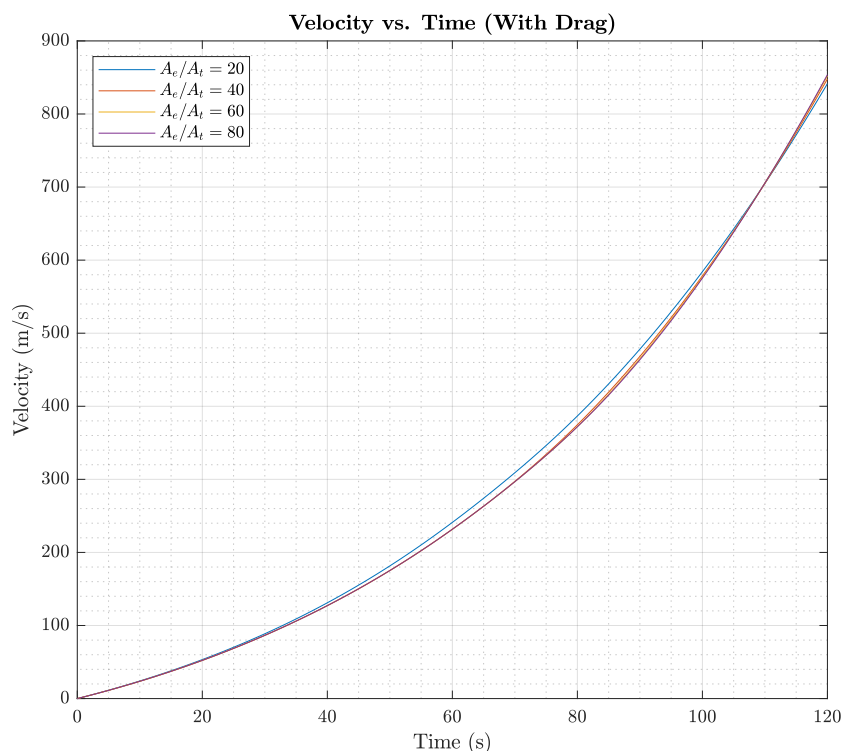
Variable	$A_e/A_t = 20$	$A_e/A_t = 40$	$A_e/A_t = 60$	$A_e/A_t = 80$
Alçada (km)	37.105	36.450	36.403	36.381
Velocitat (m/s)	867.587	874.222	878.531	880.651
Mach	2.794	2.815	2.829	2.836

**Taula 5.1** Valors de les variables finals alçada, velocitat i Mach transcorreguts 120 segons des de l'enlairament, sense considerar el Drag

## 5.2 Resultats amb resistència aerodinàmica



**Figura 5.3** Evolució de la variable alçada durant la trajectòria d'enlairament considerant el Drag



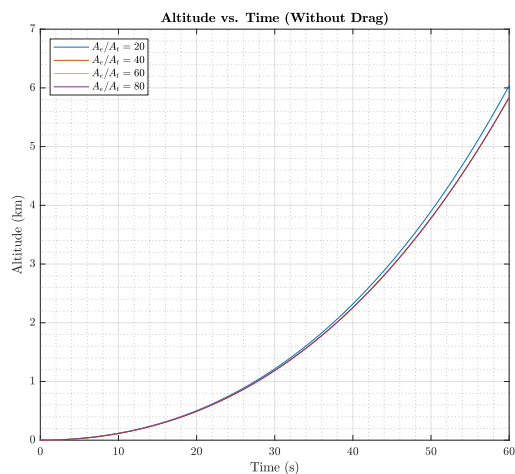
**Figura 5.4** Evolució de la variable velocitat durant la trajectòria d'enlairament considerant el Drag

Per al en que es té en compte la resistència aerodinàmica novament observem com la evolució de les dues principals variables que han estat objecte d'estudi presenten tendències molt similars al cas anterior. No obstant, la diferència respecte a les gràfiques en les que es mostren els resultats menyspreant l'efecte d'aquestes forces dissipatives s'aprecia en els valors finals d'alçada i velocitat que el coet es capaç d'assolir. Es cert que no hi ha massa diferència entre uns valors i els altres, fet que es pot justificar si es té en compte que la força aerodinàmica resultarà ser una força de mòdul molt més petit que les forces pes o empenyiment i com a conseqüència la seva contribució també serà més reduïda. Tot i que la diferència es petita, considerem bo estudiar aquests resultats i comparar-los amb el cas ideal, sense resistència per tal de treure conclusions que s'apropin el màxim a la realitat d'aquests llançaments, sempre dins de les hipòtesis inicials que s'han considerat.

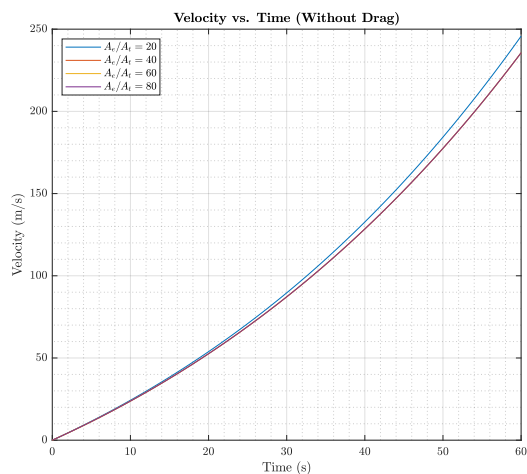
Així mateix de la mateixa manera que s'han presentat els resultats finals en l'apartat anterior, per a aquest cas es té:

Variables finals	$A_e/A_t = 20$	$A_e/A_t = 40$	$A_e/A_t = 60$	$A_e/A_t = 80$
Alçada (km)	35.937	35.335	35.283	35.258
Velocitat (m/s)	841.510	848.076	852.017	853.870
Mach	2.727	2.748	2.761	2.767

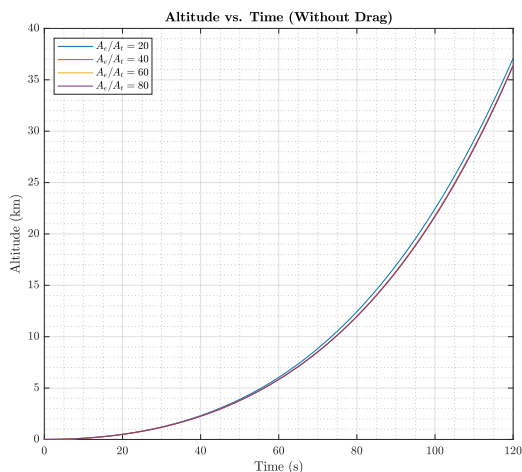
**Taula 5.2** Valors de les variables finals alçada, velocitat i *Mach* transcorreguts 120 segons des de l'enlairament, considerant el *Drag*



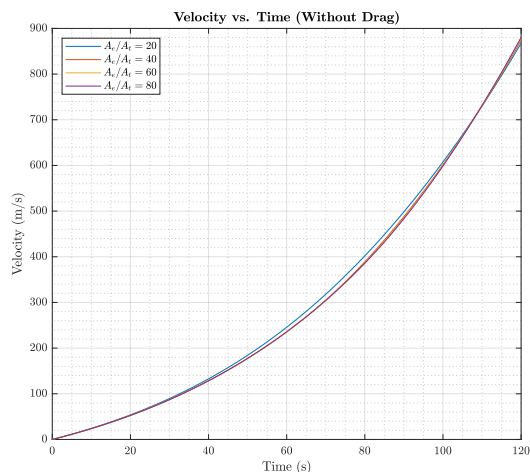
(a) Increment d'alçada al cap de 60 segons



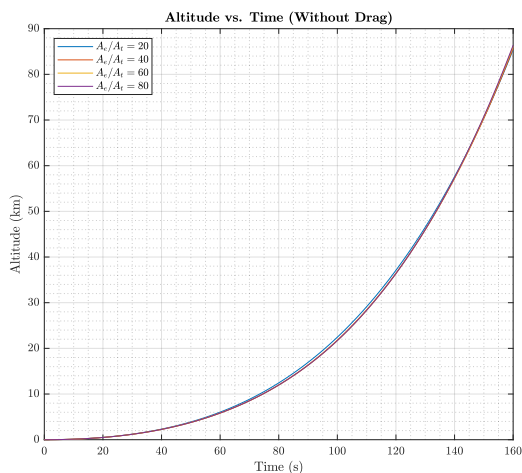
(b) Increment de velocitat al cap de 60 segons



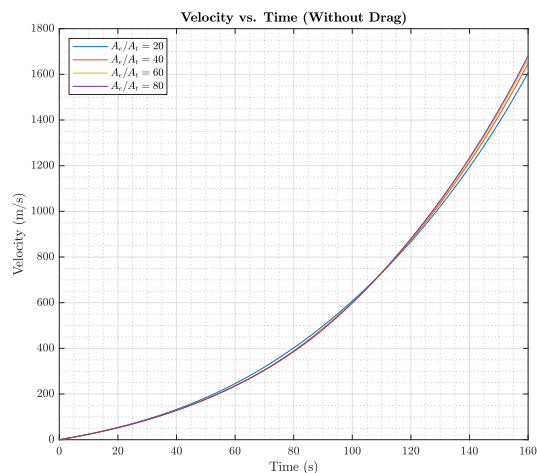
(c) Increment d'alçada al cap de 120 segons



(d) Increment de velocitat al cap de 120 segons



(e) Increment d'alçada al cap de 160 segons



(f) Increment de velocitat al cap de 160 segons

**Figura 5.5** Evolució de alçada i velocitat del coet, per al cas en que es negligeix la resistència aerodinàmica

A continuació es presenten els valors finals de les variables alçada, velocitat i número de Mach, per a les diferents relacions d'àrees i temps d'encesa que s'han considerat, en tots elles sense computar els efectes de la resistència aerodinàmica.

Temps (s)	$A_e/A_t = 20$	$A_e/A_t = 40$	$A_e/A_t = 60$	$A_e/A_t = 80$
60	6.049	5.839	5.839	5.839
120	37.105	36.450	36.403	36.381
160	85.436	85.684	86.122	86.382

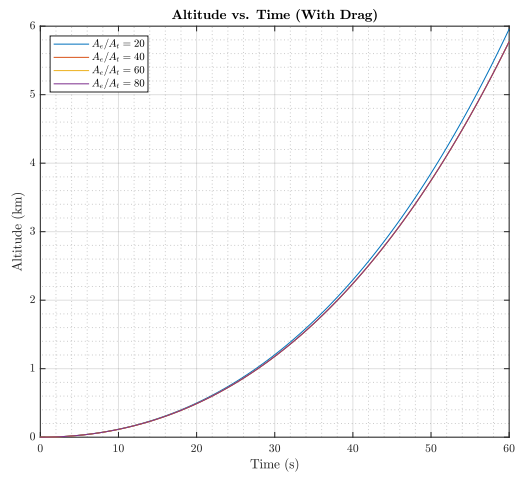
**Taula 5.3** Valors de la variable alçada en kilòmetres transcorreguts 60-120 i 160 segons des de l'enlairament, menyspreant el *Drag*

Temps (s)	$A_e/A_t = 20$	$A_e/A_t = 40$	$A_e/A_t = 60$	$A_e/A_t = 80$
60	245.742	235.649	235.649	235.649
120	867.587	874.222	878.531	880.651
160	1607.434	1647.808	1668.631	1681.250

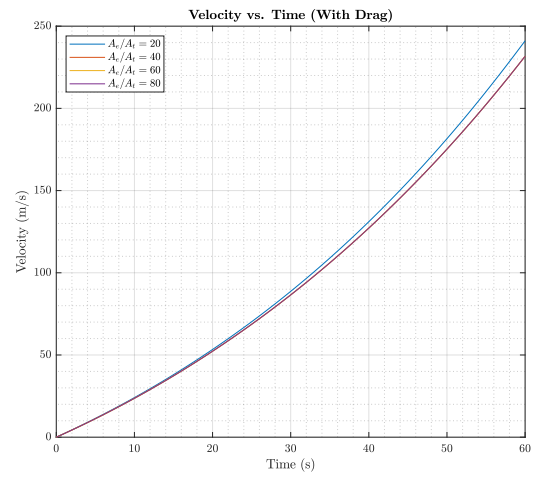
**Taula 5.4** Valors de la variable velocitat en metres/segon transcorreguts 60-120 i 160 segons des de l'enlairament, menyspreant el *Drag*

Temps (s)	$A_e/A_t = 20$	$A_e/A_t = 40$	$A_e/A_t = 60$	$A_e/A_t = 80$
60	0.777	0.745	0.745	0.745
120	2.794	2.815	2.829	2.836
160	5.866	6.013	6.089	6.135

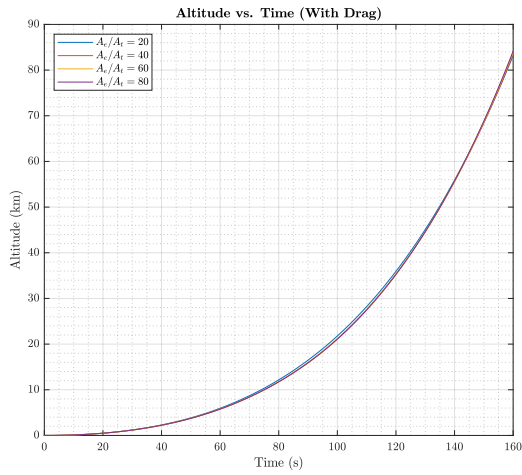
**Taula 5.5** Valors del número de Mach (adimensional) transcorreguts 60-120 i 160 segons des de l'enlairament, menyspreant el *Drag*



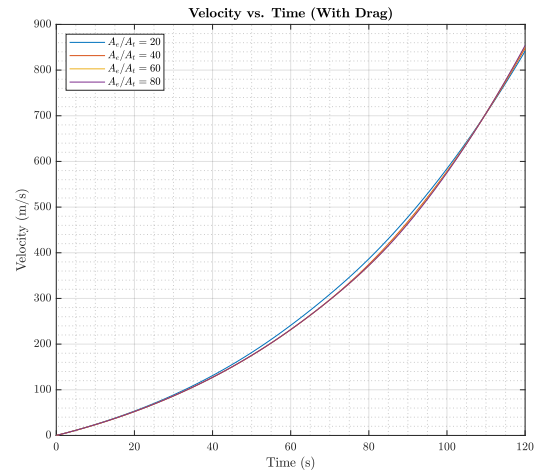
(a) Increment d'alçada al cap de 60 segons



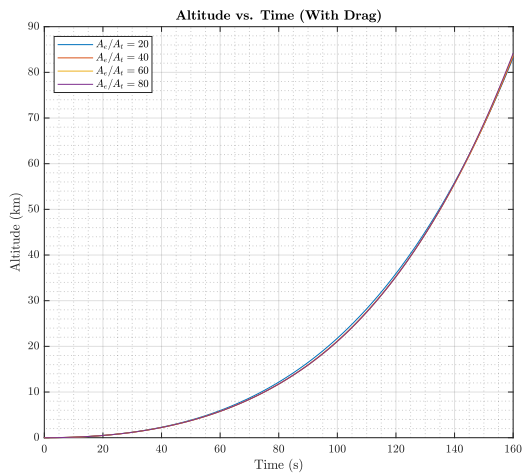
(b) Increment de velocitat al cap de 60 segons



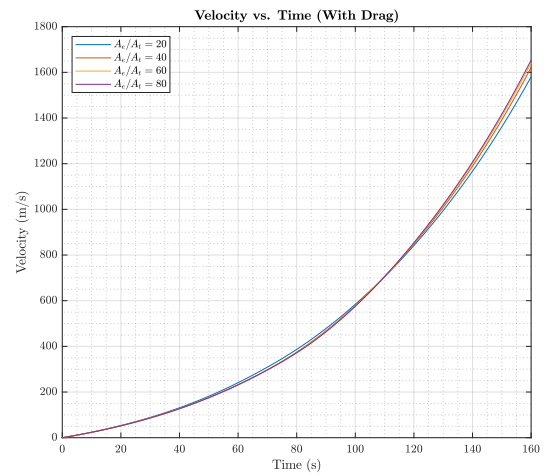
(c) Increment d'alçada al cap de 120 segons



(d) Increment de velocitat al cap de 120 segons



(e) Increment d'alçada al cap de 160 segons



(f) Increment de velocitat al cap de 160 segons

**Figura 5.6** Evolució de alçada i velocitat del coet, per al cas en que es considera la resistència aerodinàmica

Amb l'objectiu de presentar els resultats d'una forma clara i entenedora alhora que ens permeten analitzar i extreure conclusions sobre la trajectòria d'ascens del coet, es presenta a continuació una taula amb els valors finals de alçada, velocitat i número de Mach en funció de les diferents relacions d'àrees i els temps d'encesa que han estat objecte d'estudi durant tot el projecte, per al cas d'estudi més realista en el que s'ha considerat la contribució de la resistència aerodinàmica.

Temps (s)	$A_e/A_t = 20$	$A_e/A_t = 40$	$A_e/A_t = 60$	$A_e/A_t = 80$
60	5.958	5.775	5.775	5.775
120	35.937	35.335	35.283	35.258
160	83.208	83.505	83.922	84.167

**Taula 5.6** Valors de la variable alçada en kilòmetres transcorreguts 60-120 i 160 segons des de l'enlairament, considerant el *Drag*

Temps (s)	$A_e/A_t = 20$	$A_e/A_t = 40$	$A_e/A_t = 60$	$A_e/A_t = 80$
60	241.112	231.712	231.712	231.712
120	841.510	848.076	852.017	853.870
160	1580.833	1621.066	1641.493	1653.820

**Taula 5.7** Valors de la variable velocitat en metres/segon transcorreguts 60-120 i 160 segons des de l'enlairament, considerant el *Drag*

Temps (s)	$A_e/A_t = 20$	$A_e/A_t = 40$	$A_e/A_t = 60$	$A_e/A_t = 80$
60	0.755	0.725	0.725	0.725
120	2.727	2.748	2.761	2.767
160	5.769	5.916	5.990	6.035

**Taula 5.8** Valors del número de Mach (adimensional) transcorreguts 60-120 i 160 segons des de l'enlairament, considerant el *Drag*



## 6 Relació d'àrees òptima

Una vegada analitzades les diferents evolucions de l'alçada i la velocitat per a relacions d'àrea equivalents a  $A_e/A_t = 20, 40, 60, 80$ , s'observen els resultats finals d'aquestes variables per als diferents temps d'encesa. A continuació s'extreuen tota una sèrie de conclusions que permeten considerar quina relació d'àrees resultaria la millor elecció a l'hora de dissenyar la tovera del coet.

- S'ha observat que com el temps d'encesa del motor juga un paper crucial a l'hora de determinar els valors finals d'alçada, velocitat i consegüentment número de Mach.
- Per a temps d'encesa notablement baixos com 60 segons, clarament s'aprecia com grans relacions de  $\frac{A_e}{A_t}$  no resulten beneficioses. Això és degut a que durant els primers instants de l'enlairament la pressió atmosfèrica té valors encara massa grans com per impedir que toveres amb relacions  $A_e/A_t$  de 40, 60 i 80 presentin gran part de la capa límit del seu flux despresa i com a conseqüència els gasos no puguin expandir-se suficientment i generar tot l'empenyiment per al qual han estat dissenyades. Això provoca que els resultats per a relacions de  $A_e/A_t \approx 20$  siguin considerablement superiors en comparació amb els altres en aquesta situació.
- Si es consideren temps d'encesa en que aquesta primera etapa ha de permetre ascendir al coet fins a alçades majors, els resultats presenten certes variacions respecte als anteriors. Per a temps de funcionament mitjos com per exemple entorn de 120 segons, el comportament del coet és més proper a l'esperat per a relacions d'àrees més grans. No obstant això, la màxima alçada del vehicle es continua assolint per la relació d'àrees més petit, és a dir,  $A_e/A_t = 20$ . Pel que fa a la velocitat, el major valor es dona per la màxima relació d'àrees estudiada.
- Per últim, atenent als últims llançaments, com per exemple el del coet *Falcon 9*, la primera etapa té un temps de funcionament fins als 160 segons després del llançament. Per aquest motiu, s'ha triat aquest temps pel tercer cas d'estudi. En aquest cas, els resultats obtinguts denoten la superioritat de les toveres amb grans relacions d'àrees a mesura que la pressió atmosfèrica disminueix com conseqüència de l'augment de l'alçada. D'aquesta manera es pot aprofitar el gran potencial d'aquestes a l'hora de la generació d'empenyiment i motiu pel qual han estat dissenyades.

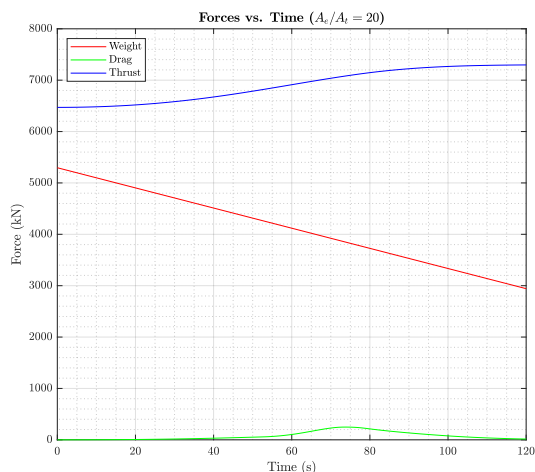
D'aquesta manera, es pot concloure que la clau per decidir el disseny de tovera adequat per a cada cas depèn en gran part del temps d'encesa o vida útil de l'etapa llançadora que la missió requereixi. Si es tracta d'una missió amb una primera etapa amb un temps de funcionament d'aproximadament 1 minut, la relació d'àrees més favorable  $A_e/A_t \approx 20$ . Tal com es mostra en les taules 5.2, aquesta és la relació que permet assolir la major alçada així com major increment de velocitat. Així, per a baixos valors de temps d'encesa, l'ús de relacions  $A_e/A_t$  superiors no compensa en cap moment la complexitat constructiva ni el cost econòmic.

D'altra banda, si és precís que aquesta primera etapa sigui capaç de d'eleva el coet a alçades majors i generar impulsos de l'ordre 1000 m/s, convindrà treballar amb relacions d'àrees majors com  $A_e/A_t = 60$  o 80. En aquest cas, el guany d'empenyiment que d'aquest tipus de toveres quan el vehicle es troba a alçades considerables, sí compensa les dificultats addicionals que suposa utilitzar-les.

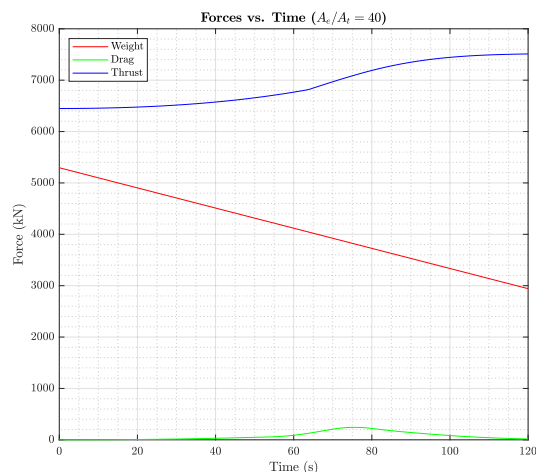
Tanmateix, com s'ha decidit centrar la major part de l'estudi en l'evolució de les variables durant els dos primers minuts de vol, es considerarà la millor relació d'àrees en funció dels resultats obtinguts per a  $t_b = 120$  s. Tal com s'ha comentat, els resultats obtinguts presenten certa contradicció, ja que majors relacions d'àrees permetran obtenir impulsos de velocitat més grans, alhora que menors relacions  $A_e/A_t$  permetran assolir alçades superiors una vegada transcorreguts aquests dos minuts. Per decidir s'ha de triar un criteri d'elecció que permeti prioritzar entre guany d'impuls (major velocitat final) o un major increment d'alçada. Tot i així, pel cas estudiat, els valors d'alçada finals entre  $A_e/A_t = 20$  i  $A_e/A_t = 80$  difereixen aproximadament d'uns 680 m. D'aquest mode, tal com s'ha mencionat en el plantejament del projecte, al disposar el nostre coet de més d'una etapa, es dona per suposat que aquest continuarà ascendint una vegada es desprengui aquesta primera i aquesta variació d'alçada perdrà importància en front d'un major increment de velocitat que proporcionaria la tovera amb  $\frac{A_e}{A_t} = 80$ .

Finalment, com a conclusió global, es pot que si no es tenen en compte els efectes de pes addicional que suposa tenir una tovera amb major diàmetre de sortida, la relació entre l'àrea de sortida i l'àrea de la gola de la tovera  $A_e/A_t = 80$  proporcionarà els resultats més òptims a l'enlairament del coet.

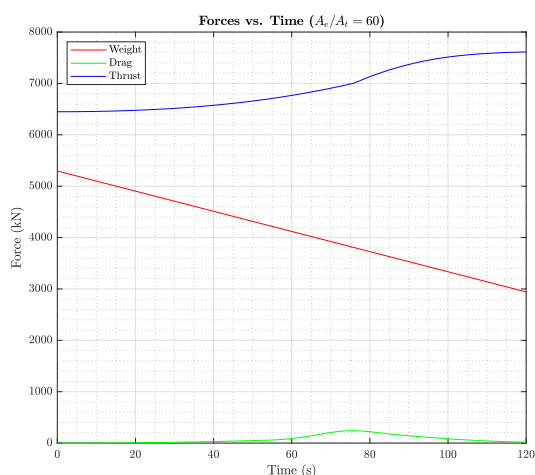
## 7 Evolució de les forces durant la trajectòria



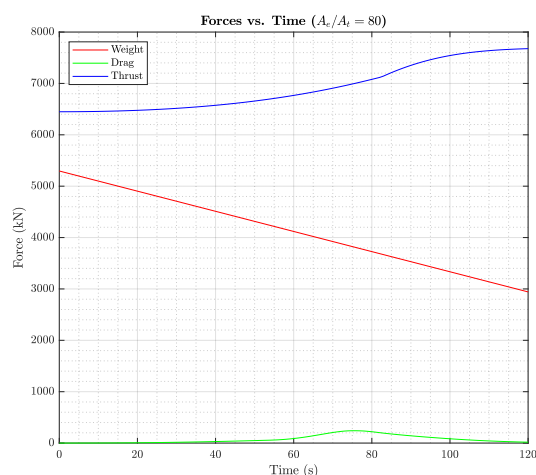
(a) Evolució de les forces per  $A_e/A_t = 20$



(b) Evolució de les forces per  $A_e/A_t = 40$



(c) Evolució de les forces per  $A_e/A_t = 60$



(d) Evolució de les forces per  $A_e/A_t = 80$

**Figura 7.1** Evolució de les forces de drag, empenya i pes per les relacions d'àrees  $A_e/A_t = 20, 40, 60, 80$

En la Figura 7.1 es representen l'evolució temporal de les forces d'empenyiment (*Thrust*), resistència aerodinàmica (*Drag*) i del pes del coet (*Weight*) per diferents relacions d'àrea.

Aquestes forces són les diferents contribucions que apareixen a l'equació del moviment. En efecte, la contribució més gran que apareix en totes les gràfiques és l'empenyiment. Es pot observar que a mesura que s'augmenta la relació d'àrees l'empenyiment augmenta de forma progressiva on el màxim es troba al empenyiment per a  $t = 120$  s.

El pes del coet pren una forma lineal, això s'explica perquè l'expulsió de massa per unitat de temps és constant i per tant, el pes disminueix linealment. El punt inicial correspon precisament al pes inicial de l'avió 540 t i amb un flux màssic de  $\dot{m} = 2000$  kg/s, independentment de la la relació d'àrea el pes final del coet és d'aproximadament 3 t.

D'altra banda, la força *Drag* és relativament petita. És una contribució bastant petita enfront el *Thrust*. Enfront l'enorme pes del coet, en aquest cas en particular, també s'observa que es pot negligir l'efecte de la resistència aerodinàmica doncs el seu efecte està molt per sota de les altres forces.

Així doncs, l'anàlisi plantejat en aquest projecte, distingint el cas de l'estudi de l'ascens vertical del coet sense drag i amb drag. Com a conclusió, els resultats de les corbes altitud i velocitat són molt similars i ja s'intuïa a primera impressió que l'efecte del drag era possiblement negligible. Aquests resultats, verifiquen aquesta intuïció inicial i l'error que es cometria negligint el *Drag* seria mínim.

A més, com a comentari sobre l'impuls específic  $I_{sp}$  que es regeix per la següent expressió:

$$I_{sp} = \frac{F}{\dot{m}g} \quad (7.1)$$

on  $F$  és l'empenyiment,  $\dot{m}$  és el flux màssic i  $g = 9.81 \text{ m/s}^2$ .

Per tant, com el flux màssic és constant i la  $g$  és  $9.81 \text{ m/s}$  per definició. La tendència que pren el  $I_{sp}$  són idèntics però reduït en un factor de  $2000 \cdot 9.81 = 19620$ .

A continuació en la Taula 7.1 es mostren els valors màxims de les contribucions de les forces.

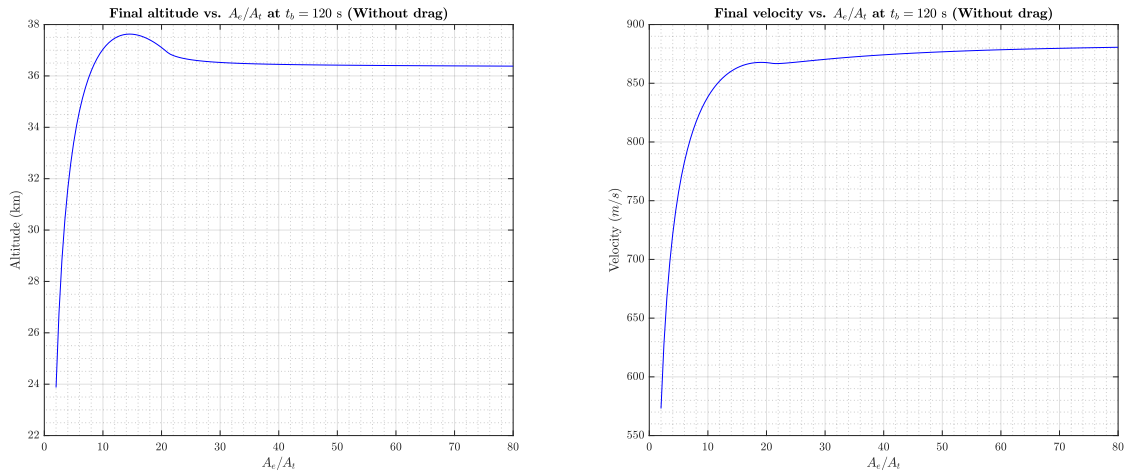
$A_e/A_t$	<i>Thrust</i> (kN)	$I_{sp}$ (s)	<i>Drag</i> (kN)
20	7297	372	248
40	7511	383	242
60	7615	388	240
80	7679	392	240

**Taula 7.1** Valors màxims de l'empenyiment (*Thrust*), la resistència aerodinàmica (*Drag*) i impuls específic ( $I_{sp}$ ) per diferents relacions d'àrea

Els valors del  $I_{sp}$  màxims de l'impuls específic són similars degut que l'empenyiment produïda pel *Thrust* en totes les relacions d'àrea són pràcticament de la mateixa magnitud. Aquest impuls específic augmenta perquè cada cop hi ha més *Thrust* i perquè hi ha menys pressió atmosfèrica. Existeix un punt en què varia la tendència de la corba, que és el punt en la qual passa de ser una tovera adaptada a una tovera sobre-expansionada.

## 8 Alçada i velocitats finals front a la relació d'àrees

Seguidament és interessant estudiar per a cada relació d'àrees l'alçada i la velocitat final que s'assoliria:



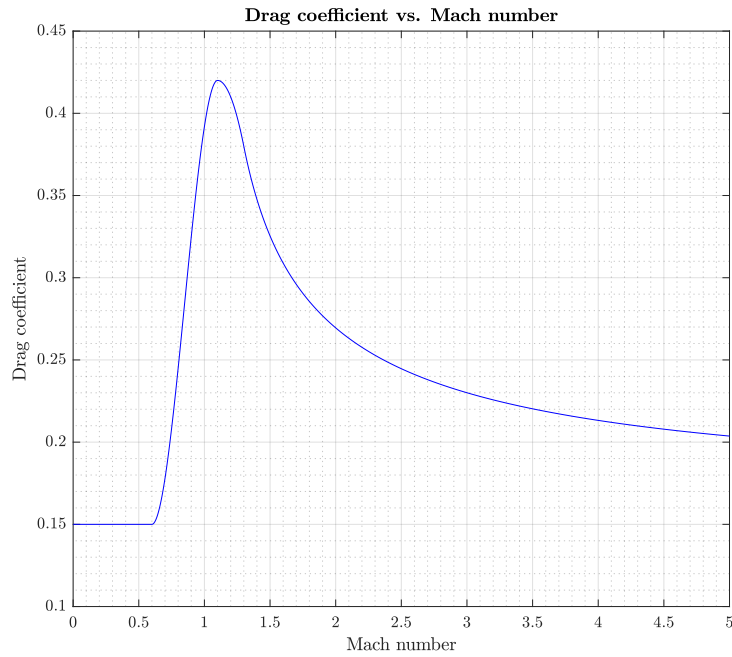
**Figura 8.1** Velocitat final en funció de  $A_e/A_t$  (2 – 80) per  $t_b = 120$  s, sense considerar el drag

La màxima alçada es produeix al voltant de 37.5 km, per una relació per a  $A_e/A_t = 15$ . I a partir d'aquí l'alçada final es redueix. És important destacar com aquest valor s'estabilitza a 36.5 km.

La gràfica de la velocitat final és important ja que a partir de la velocitat final, el coet ja augmenta d'alçada per inèrcia. De la mateixa forma, el pic de màxima velocitat es troba al voltant dels 15 km i s'estabilitza a uns valors de 875 m/s per al relacions d'àrea superiors a 20.

## A Coeficient de resistència aerodinàmica

$$C_D = \begin{cases} 0.15 & \text{si } M \leq 0.6 \\ -4.32M^3 + 11.016M^2 - 8.5536M + 2.24952 & \text{si } 0.6 < M \leq 1.1 \\ -M^2 + 2.2M - 0.79 & \text{si } 1.1 < M \leq 1.3 \\ 0.16769 + \frac{0.17636}{\sqrt{M^2 - 1}} & \text{si } 1.3 < M \leq 5.0 \end{cases}$$



**Figura A.1** Dependència del coeficient de resistència aerodinàmica ( $C_D$ ) amb el número de Mach ( $M$ )

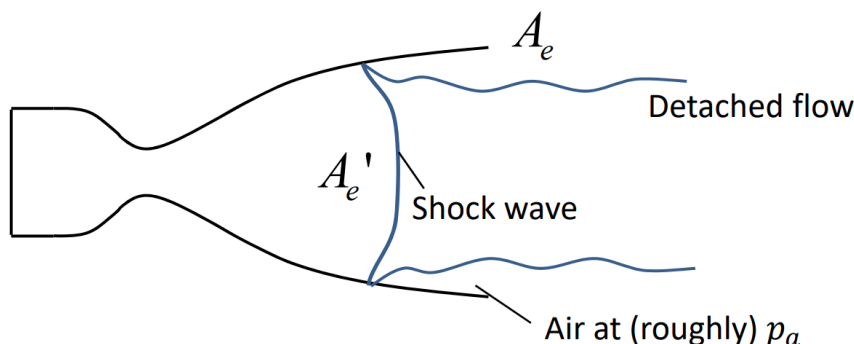
Clarament es pot observar com l'efecte de la resistència aerodinàmica es veurà notablement amplificat quan s'estigui volant a velocitats properes a la del so, entre  $Mach = 0.9$  i  $Mach = 1.2$  aproximadament. Aquesta distribució de coeficient de *Drag* denota clarament com per valors de velocitat en els que el vehicle voli en condicions de règim subsònic incompressible la resistència aerodinàmica prendrà valors molt petits fins al punt de que es podrà menysprear.

## B Descripció del Criteri de Summerfield

En toveres sobre-expansionades, es produeix una contracció del flux. Aquesta contracció és precisament la que causa aquesta ona de xoc obliqua. Aquesta ona de xoc obliqua obligarà a canviar la direcció de l'aire i el moment en què es trobin totes les ones de xoc al llarg de la circumferència de la tovera, és quan es produirà l'ona de xoc normal. Aigües avall, s'anirà produint una sèrie d'expansions i contraccions. Aquest fenomen es coneix con la **configuració dels diamants**.

Per toveres lleugerament sobre expansionades, la pressió després el çorro", és igual a la pressió ambient. En canvi, per toveres cada cop més sobre expansionades, l'angle de l'ona de xoc serà cada cop més gran fins arribar un moment en què l'ona de xoc entra a dins de la tovera.

En cas que la tovera sigui altament sobre-expansionada, l'ona de xoc es va desplaçant aigües amunt fins i tot arribant a entrar a dintre de la tovera. L'ona de xoc produeix un canvi de pressió molt brusc i sobtat i com a conseqüència, es desprèn la capa límit. Entre la zona del flux després i la paret de la tovera, es produeix una forta succió on l'aire que és succionada es troba virtualment a pressió ambient  $p_a$ .



**Figura B.1** Esquema del flux en una tovera sobre-expansionada (vegeu la ona de xoc normal i el flux després). A partir de [3].

En definitiva, la força neta sobre les cares de la paret de la tovera és nul·la. Per tant, el tros de tovera per sota de l'ona de xoc és com si no existís. Empíricament, s'ha observat que aquest fenomen no és perjudicial sinó que tot el contrari. Aquest fet del despreniment de la capa límit, permet que les pressions entre la paret de la tovera i l'exterior sigui gairebé iguals i la succió no pren massa força.

Amb el *criteri de Summerfield*, matemàticament no té en compte la part de tovera sobre expansionada i per tant s'ha de considerar una àrea efectiva.

Com a il·lustració d'aquest fenomen, vegeu la primera il·lustració de [1].

## C Integració numèrica Runge-Kutta RK4

### C.1 Resolució numèrica de sistemes d'equacions diferencials

Es considera un problema de valor inicial com el següent:

$$\begin{cases} \frac{dy}{dx} = f(x, y(x)) \\ y(a) = y_0 \end{cases} \quad \begin{matrix} (C.1) \\ (C.2) \end{matrix}$$

amb  $x \in [a, b]$ ,  $y(x) \in \mathbb{R}$  i  $f: [a, b] \times \mathbb{R} \rightarrow \mathbb{R}$ . En general, la major part d'equacions diferencials que es plantegen no tenen una solució explícita  $y(x)$ . No obstant, donats punts  $x_0, \dots, x_n \in [a, b]$ , es pot obtenir el valor que pren  $y(x)$  en aquests [2].

Els algorismes de Runge-Kutta venen donats per [2]:

$$\begin{cases} y(a) = y_0 \\ y_{n+1} = y_n + h \sum_{i=1}^k c_i k_i^n, \quad 0 \leq n \leq N-1 \end{cases} \quad \begin{matrix} (C.3) \\ (C.4) \end{matrix}$$

amb  $k \in \mathbb{N}$  definit prèviament, i els coeficients  $k_i^n$  són funcions definides per:

$$\begin{cases} k_1^n = f(x_n, y_n) \\ k_i^n = f\left(x_n + a_i h, y_n + h \sum_{j=1}^{i-1} b_{ij} k_j^n\right), \quad 2 \leq i \leq k \end{cases} \quad \begin{matrix} (C.5) \\ (C.6) \end{matrix}$$

L'algorisme RK4, és l'algorisme de Runge-Kutta que s'obté per  $k = 4$ . Aquest s'escriu com:

$$\begin{cases} y(a) = y_0 \\ y_{n+1} = y_n + \frac{h}{6} (k_1^n + 2k_2^n + 2k_3^n + k_4^n), \quad 0 \leq n \leq N-1 \end{cases} \quad \begin{matrix} (C.7) \\ (C.8) \end{matrix}$$

on els coeficients es defineixen per:

$$\begin{cases} k_1^n = f(x_n, y_n) \\ k_2^n = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} k_1^n\right) \end{cases} \quad \begin{matrix} (C.9) \\ (C.10) \end{matrix}$$

$$\begin{cases} k_3^n = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} k_2^n\right) \\ k_4^n = f(x_n + h, y_n + h k_3^n) \end{cases} \quad \begin{matrix} (C.11) \\ (C.12) \end{matrix}$$

Intuïtivament, els mètodes de Runge-Kutta discretitzen el domini definit per la variable independent  $x$  en passes definides per  $h$ , i per a cada punt  $x_n$  de la discretització, calculen el valor de  $y'$  en punts de l'entorn de  $x_n$ , i.e., els coeficients  $k_i^n$ . A partir del valor actual  $y_n$  de la funció i dels coeficients, es calcula el següent valor de la funció,  $y_{n+1}$ .

Els mètodes numèrics són, en general, aproximacions a la solució analítica del problema. La precisió del l'aproximació depèn de  $h$ . No obstant, el mètode RK4 és d'ordre global 4, és a dir, l'error és de l'ordre de  $\mathcal{O}(h^4)$ . Això implica que, si es defineix  $h' = h/2$  i s'aplica novament l'algorisme RK4, l'error es reduirà 16 vegades.

Els mètodes de Runge-Kutta i, en particular el mètode RK4, són fàcilment generalitzables a sistemes d'equacions diferencials ordinàries, com es veurà a continuació.



## C.2 Resolució de la trajectòria sense resistència aerodinàmica

El sistema d'equacions diferencials de la trajectòria sense resistència aerodinàmica és el següent:

$$\begin{cases} \frac{dh}{dt} = V \\ \frac{dV}{dt} = \frac{\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} \frac{\gamma M + 1/M}{\sqrt{1 + \frac{\gamma-1}{2} M^2}} - \frac{p}{p_c} \frac{A_e}{A_t}}{m_0 - \dot{m}t} - g \end{cases} \quad (\text{C.13})$$

Per simplicitat, cada equació diferencial del sistema es pot expressar com una funció del temps  $t$ , velocitat  $V$  i alçada  $h$ :

$$\begin{cases} F_1(t, V, h) = \frac{dh}{dt} \\ F_2(t, V, h) = \frac{dV}{dt} \end{cases} \quad (\text{C.15})$$

Els coeficients de l'algorisme de Runge-Kutta són els següents:

$$j_1^n = F_1(t_n, V_n, h_n) \quad (\text{C.17})$$

$$k_1^n = F_2(t_n, V_n, h_n) \quad (\text{C.18})$$

$$j_2^n = F_1\left(t_n + \frac{\Delta t}{2}, V_n + \frac{\Delta t}{2} j_1^n, h_n + \frac{\Delta t}{2} k_1^n\right) \quad (\text{C.19})$$

$$k_2^n = F_2\left(t_n + \frac{\Delta t}{2}, V_n + \frac{\Delta t}{2} j_1^n, h_n + \frac{\Delta t}{2} k_1^n\right) \quad (\text{C.20})$$

$$j_3^n = F_1\left(t_n + \frac{\Delta t}{2}, V_n + \frac{\Delta t}{2} j_2^n, h_n + \frac{\Delta t}{2} k_2^n\right) \quad (\text{C.21})$$

$$k_3^n = F_2\left(t_n + \frac{\Delta t}{2}, V_n + \frac{\Delta t}{2} j_2^n, h_n + \frac{\Delta t}{2} k_2^n\right) \quad (\text{C.22})$$

$$j_4^n = F_1(t_n + \Delta t, V_n + \Delta t j_3^n, h_n + \Delta t k_3^n) \quad (\text{C.23})$$

$$k_4^n = F_2(t_n + \Delta t, V_n + \Delta t j_3^n, h_n + \Delta t k_3^n) \quad (\text{C.24})$$

Finalment, les variables en el temps  $t_{n+1}$  es calculen com:

$$t_{n+1} = t_n + \Delta t \quad (\text{C.25})$$

$$V_{n+1} = V_n + \frac{\Delta t}{6} (j_1^n + 2j_2^n + 2j_3^n + j_4^n) \quad (\text{C.26})$$

$$h_{n+1} = h_n + \frac{\Delta t}{6} (k_1^n + 2k_2^n + 2k_3^n + k_4^n) \quad (\text{C.27})$$

Com es pot apreciar, (C.13) no és funció de  $t$  o  $h$ , com sembla indicar  $F_1$ . No obstant, per motius de completesa, quan es planteja l'algorisme de Runge-Kutta s'acostuma a escriure que cada funció derivada  $F_1, \dots, F_n$  depèn de totes les variables  $x_1, \dots, x_{n+1}$ . Tanmateix, en el codi de MATLAB adjunt només es posa les variables de les quals cada derivada depèn.

### C.3 Resolució de la trajectòria amb resistència aerodinàmica

El sistema d'equacions diferencials de la trajectòria amb resistència aerodinàmica es el següent:

$$\left\{ \begin{array}{l} \frac{dh}{dt} = V \\ \frac{dV}{dt} = \frac{\left( \frac{2}{\gamma+1} \right)^{\frac{\gamma+1}{2(\gamma-1)}} \frac{\gamma M + 1/M}{\sqrt{1 + \frac{\gamma-1}{2} M^2}} - \frac{p}{p_c} \frac{A_e}{A_t} - \frac{1}{2} \rho V^2 S_{ref} C_D}{m_0 - \dot{m}t} - g \end{array} \right. \quad (C.28)$$

(C.29)

L'esquema de l'algorisme de Runge-Kutta és idèntic al presentat en l'anterior secció.

## D Codi del projecte

Es presenta a continuació el codi empleat per la resolució del projecte. Vegeu que per la resolució del sistema d'equacions diferencials s'ha fet servir el mètode de Runge-Kutta C.

## D.1 Funció pel càlcul de les temperatures de cada capa atmosfèrica

```

1 function [Tb, Pb] = getBaseTemperaturePressure(R, g0, T0, P0, Mm, H_layer, lambda)
2 %-----
3 % Inputs:
4 % - R Universal Constant for Ideal Gases [J/mole*K]
5 % - g0 Acceleration at planet's surface [m/s^2]
6 % - T0 Standard Temperature at planet's surface [K]
7 % - P0 Standard Pressure at planet's surface [Pa]
8 % - Mm Atmospheric gas molecular mass [kg*mole^-1]
9 % - H_layer Atmospheric layers altitude [m]
10 % - lambda Atmospheric layers thermal gradients [K/m]
11 %-----
12 % Outputs:
13 % - Tb Vector of base temperatures, 1 x length(H_layer) [K]
14 % - Pb Vector of base pressures, 1 x length(H_layer) [Pa]
15 % Base (magnitude) refers to the value of (magnitude( at the beginning of
16 % an atmospheric layer
17 %-----
18
19 % Declare vectors of Base Temperature and Pressure
20 Tb = zeros(1, length(H_layer)); % Base Temperature [K]
21 Pb = zeros(1, length(H_layer)); % Base Pressure [Pa]
22
23 % Compute Base Temperatures and Pressures for each layer
24 Tb(1) = T0;
25 Pb(1) = P0;
26 for i = 2:length(H_layer)
27     % Compute Base Temperature at layer i
28     Tb(i) = Tb(i-1) + lambda(i-1)*(H_layer(i)-H_layer(i-1));
29     % Compute Base Pressure at layer i
30     if lambda(i-1) == 0 % Isothermal layer
31         Pb(i) = Pb(i-1)*exp(-g0*Mm*(H_layer(i)-H_layer(i-1))/(R*Tb(i-1)));
32     else % Non-isothermal layer
33         Pb(i) = Pb(i-1)*((Tb(i-1)/(Tb(i-1)+lambda(i-1)* ...
34             (H_layer(i)-H_layer(i-1))))^(g0*Mm/(R*lambda(i-1))));
35     end
36 end
37
38 end

```

## D.2 Funció pel càlcul de les condicions atmosfèriques

```

1 function [T, P, rho, c] = computeAtmosphere(Tb, Pb, H_layer, lambda, R, g0, Mm, H)
2 %-----
3 % Inputs:
4 % - Tb Vector of base temperatures, 1 x length(H_layer) [K]
5 % - Pb Vector of base pressure, 1 x length(H_layer) [Pa]
6 % - H_layer Atmospheric layers altitude [m]
7 % - lambda Atmospheric layers thermal gradients [K/m]
8 % - gamma Ratio of specific heats [adim]
9 % - R Universal gas constant [J/(kg K)]
10 % - g0 Acceleration at planet's surface [m/s^2]
11 % - Mm Planet's air molar mass [kg/mol]
12 % - H Altitude [m]
13 %-----
14 % Outputs:
15 % - T Temperature at altitude H [K]
16 % - P Pressure at altitude H [Pa]
17 % - rho Density at altitude H [kg/m^3]
18 % - c Speed of sound at altitude H [m/s]
19 %-----
20
21 % Initialize variables
22 T = 0;
23 P = 0;
24 rho = 0;
25 c = 0;
26
27 % Find atmospheric layer
28 found = 0;
29 layer = 1;
30 while (layer <= length(H_layer)-1) && (found == 0)
31     if (H_layer(layer) <= H) && (H < H_layer(layer+1))
32         found = 1;
33     else
34         layer = layer + 1;
35     end
36 end
37
38 % Compute properties
39 if found == 1

```

```
40 % Temperature
41 T = Tb(layer) + lambda(layer)*(H - H_layer(layer));
42 % Pressure
43 if lambda(layer) == 0
44     P = Pb(layer)*exp(-g0*Mm*(H-H_layer(layer))/(R*Tb(layer)));
45 else
46     P = Pb(layer)*((Tb(layer)/(Tb(layer) + lambda(layer)*(H-H_layer(layer))))^(g0*Mm/(R*lambda(layer))));
47 end
48 % Density
49 rho = P/((R/Mm)*T);
50 % Specific heats and ratio of specific heats at constant pressure
51 cp = 1034.09 - 2.849e-1*T + 7.817e-4*T^2 - 4.971e-7*T^3 + 1.077e-10*T^4;
52 cv = cp - R/Mm;
53 gamma = cp/cv;
54 % Speed of sound
55 c = sqrt(gamma*(R/Mm)*T);
56 end
57
58 end
```

### D.3 Funció pel càlcul del *MFP*

```
1 function [Mach_menor,Mach_mayor] = MFP_function(gamma,MFP)
2     %Devuelve dos valores de Mach, el valor pequeno y el grande
3     %Para casos concretos de MFP pueden coincidir
4     Mach_menor=vpasolve(MFP == sqrt(gamma)* M / ((1 + (gamma-1)/2 *M^2)^((gamma+1)/(2*(gamma-1)))));
5     Mach_mayor=vpasolve(MFP == sqrt(gamma)* M / ((1 + (gamma-1)/2 *M^2)^((gamma+1)/(2*(gamma-1))))), M, 'Random', true);
6 end
```

### D.4 Funció per representar el *MFP*

```
1 clear;
2 close all;
3 clc;
4
5 gamma = 1.1:0.1:1.4;
6
7 M = 0:0.01:10;
8
9 MFP = zeros(length(gamma), length(M));
10 for i = 1:length(gamma)
11     MFP(i,:) = sqrt(gamma(i))*M./(1+(gamma(i)-1)/2*M.^2).^((gamma(i)+1)/(2*(gamma(i)-1)));
12 end
13
14 legend_str = cell(length(gamma),1);
15 for i = 1:length(gamma)
16     legend_str(i,1) = sprintf('$\\gamma = %.2f$', gamma(i));
17 end
18
19 figure(1);
20 hold on;
21 title('textbf{Mass flow parameter vs. Mach number}');
22 set(groot,'defaultAxesTickLabelInterpreter','latex');
23 set(groot,'defaulttextinterpreter','latex');
24 set(groot,'defaultLegendInterpreter','latex');
25 for i = 1:length(gamma)
26     plot(M, MFP(i,:));
27 end
```

```
28 xlabel("Mach number");
29 ylabel("Mass flow parameter");
30 grid on;
31 grid minor;
32 box on;
33 set(gcf, 'units', 'centimeters', 'position', [1,1,18,15]);
34 legend(legend_str);
35 hold off;
```



## D.5 Programa per evolució sense *Drag*

```

1  %% ROCKET LAUNCH
2  clear all;
3  close all;
4  clc;
5
6  %% 1. Definition of Constants, Parameters and Variables
7
8  % 1.1. CONSTANTS
9
10  G = 6.67408e-11; % Gravitational Constant [m^3/(kg*s^2)]
11  R = 8.31432; % Universal Constant for Ideal Gases [J/mole*K]
12
13  % 1.1.1. Earth
14  R_Earth = 6371.0e3; % Radius of Earth [m]
15  M_Earth = 5.9724e24; % Earth's Mass [kg]
16  g0_Earth = 9.80665; % Acceleration at Earth's surface [m/s^2]
17  T0_Earth = 288.15; % US Standard Sea Level Temperature [K]
18  P0_Earth = 101325; % Pressure at Sea Level [Pa]
19  Mm_Earth = 28.9644*10^-3; % Molecular Mass [kg*mole^-1]
20  gamma_gas_Earth = 1.4; % Earth's air specific heats relation [adim]
21
22  % Earth's atmospheric layers altitude [m]
23  H_layer_Earth = 1e3*[0 11 20 32 47 52 61 69 79 90 100 110 117.776];
24  % Earth's atmopheric layers thermal gradient [K/m]
25  lambda_layer_Earth = 1e-3*[-6.5 0 1 2.8 0 -2 -4 -3 0 2 4.36 16.4596 0];
26
27  % 2.3. Compute base temperatures and pressures
28  [Tb, Pb] = getBaseTemperaturePressure(R, g0_Earth, T0_Earth, P0_Earth, Mm_Earth, H_layer_Earth, lambda_layer_Earth);
29
30
31  % 1. 3. VARIABLES
32  syms V; % Vehicle velocity [m/s]
33  syms gamma; % Vehicle flight path angle [rad]
34  syms h; % Vehicle altitude [m]
35  syms r; % Vehicle flight range [m]
36  syms t; % Vehicle time of flight [s]
37  syms g; % Acceleration [m/s^2]
38  syms Q; % Dynamic pressure at h [Pa]
39  syms W; % Vehicle weight [N]

```

```

40 syms C_D; % Drag Coefficient [adim]
41 syms A; % Vehicle reference area used in C_D [m^2]
42 %syms beta; % Ballistic coefficient [N/m^2]
43 syms C_L; % Vehicle lift coefficient [adim]
44
45
46
47 %% Input data and Previous calculation
48
49 Pc = 100 * P0_Earth; % Pa
50 Tc = 3500; % K
51 gamma = 1.25;
52 m0 = 540e3; % kg
53 mW = 16e-3; % kg/mol
54 m_dot = 2000; % kg/sec
55 Ae_At=[20;40;60;80];
56 R_comb = R/(mW);
57 c_car = sqrt(R_comb*Tc)/0.6581; % 0.6581 from tables
58 At = (m_dot*c_car)/Pc;
59 MPF_e(:,1) = 0.6581./Ae_At(:,1);
60
61 % Solving MFP equation to obtain Mach Number
62 Me = zeros(length(MPF_e),1);
63 for i=1:length(MPF_e)
64     syms M;
65     M = vpasolve(MPF_e(i,1) == sqrt(gamma)* M / ((1 + (gamma-1)/2 *M^2)^((gamma+1)/(2*(gamma-1)))) , M, [1,6]);
66     Me(i,1) = M;
67 end
68
69 P_e = zeros(length(Ae_At),1);
70 for t=1:length(Ae_At)
71     P_e (t,1) = Pc/(1+Me(t,1)^2*((gamma-1)/2))^ (gamma/(gamma-1));
72 end
73
74 %% Numerical integration
75
76 final_time = 120; % Final time for computations [s]
77 time_step = 0.5; % Step size [s]
78 time = 0:time_step:final_time; % Calculates upto final_time
79
80 h = zeros(length(Ae_At),length(time));
81 v = zeros(length(Ae_At),length(time));
82

```

```

83 h(:,1) = 0; % initial condition of altitude
84 v(:,1) = 0; % initial condition of velocity
85
86 %t = time
87 %r = altitude
88 %s = velocity
89 %p = ambient pressure
90 %q = ambient density
91 %c = ambient speed of sound
92 %g = ambient gravity
93 %vac = coefficient thrust at vacuum
94 %m = mach exit number
95 %ratio = Area ratio
96
97 F_th = @(t,r,s) s; % Altitude EDO
98
99 F_tv = @(t,r,s,p,q,c,m,ratio) ...
    ↳ (Pc*At*((2/(gamma+1))^(gamma+1)/(2*(gamma-1))))*(gamma*m+(1/m))/(sqrt(1+(m^2*(gamma-1)/2)))-((p/Pc)*ratio))/ (m0-m_dot*t)) - ...
    ↳ g0_Earth; % Velocity EDO
100
101 for i=1:length(Ae_At)
102
103     for j=1:(length(time)-1) % calculation loop
104
105         [T, P, rho, a] = computeAtmosphere(Tb, Pb, H_layer_Earth, lambda_layer_Earth, R, g0_Earth, Mm_Earth, h(i,j));
106
107         if P_e(i,1) < (0.4*P)
108
109             Me_prima(i,1) = sqrt((2/(gamma-1)) * ((Pc/(0.4*P)) ^ ((gamma-1)/gamma)-1));
110             Ae_At_prima(i,1) = 0.6581 / (sqrt(gamma)*Me_prima(i,1)/(1+((gamma-1)/2)*Me_prima(i,1)^2 ^ ((gamma+1)/(2*(gamma-1)))));
111
112             k_1h = F_th(time(j),h(i,j),v(i,j));
113             k_1v = F_tv(time(j),h(i,j),v(i,j),P,rho,a,Me_prima(i,1),Ae_At_prima(i,1));
114
115             k_2h = F_th(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v);
116             k_2v = F_tv(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v,P,rho,a,Me_prima(i,1),Ae_At_prima(i,1));
117
118             k_3h = F_th((time(j)+0.5*time_step),(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v));
119             k_3v = F_tv((time(j)+0.5*time_step),(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v),P,rho,a,Me_prima(i,1),...
120                 Ae_At_prima(i,1));
121
122             k_4h = F_th((time(j)+time_step),(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step));
123             k_4v = F_tv((time(j)+time_step),(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step),P,rho,a,Me_prima(i,1),Ae_At_prima(i,1));

```

```

124 %Cf_v (i,1) = ((2/(gama+1))^(gama+1)/(2*(gama+1)))*(gama*Me(i,1)+(1/Me(i,1)))/(sqrt(1+(Me(i,1)^2*(gama-1)/2)));
125
126
127 else
128
129 k_1h = F_th(time(j),h(i,j),v(i,j));
130 k_1v = F_tv(time(j),h(i,j),v(i,j),P,rho,a,Me(i,1),Ae_At(i,1));
131
132 k_2h = F_th(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v);
133 k_2v = F_tv(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v,P,rho,a,Me(i,1),Ae_At(i,1));
134
135 k_3h = F_th((time(j)+0.5*time_step),(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v));
136 k_3v = F_tv((time(j)+0.5*time_step),(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v),P,rho,a,Me(i,1),Ae_At(i,1));
137
138 k_4h = F_th((time(j)+time_step),(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step));
139 k_4v = F_tv((time(j)+time_step),(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step),P,rho,a,Me(i,1),Ae_At(i,1));
140
141 end
142
143 h(i,j+1) = h(i,j) + (1/6)*(k_1h+2*k_2h+2*k_3h+k_4h)*time_step; % main altitude equation
144 v(i,j+1) = v(i,j) + (1/6)*(k_1v+2*k_2v+2*k_3v+k_4v)*time_step; % main altitude equation
145
146
147 end
148
149 end
150
151 legend_str = cell(length(Ae_At), 1);
152 for i = 1:length(Ae_At)
153     legend_str(i,1) = sprintf('$A_e/A_t = %d$', Ae_At(i));
154 end
155
156 figure(1);
157 hold on;
158 title("\textbf{Altitude vs. Time (Without Drag)}");
159 set(groot,'defaultAxesTickLabelInterpreter','latex');
160 set(groot,'defaultTextInterpreter','latex');
161 set(groot,'defaultLegendInterpreter','latex');
162 for i = 1:length(Ae_At)
163     plot(time, h(i,:)/1e3);
164 end
165 xlabel("Time $\left( \mathrm{s} \right)$");
166 ylabel("Altitude $\left( \mathrm{km} \right)$");

```

```
167 set(gcf, 'units', 'centimeters', 'position', [1, 1, 18, 15]);
168 grid on;
169 grid minor;
170 box on;
171 legend(legend_str, 'Location', 'Northwest');
172 hold off;
173
174
175 figure(2);
176 hold on;
177 title("\textbf{Velocity vs. Time (Without Drag)}")
178 set(groot, 'defaultAxesTickLabelInterpreter', 'latex');
179 set(groot, 'defaultTextInterpreter', 'latex');
180 set(groot, 'defaultLegendInterpreter', 'latex');
181 for i = 1:length(Ae_At)
182     plot(time, v(i,:));
183 end
184 xlabel("Time $\left( \mathrm{s} \right)$");
185 ylabel("Velocity $\left( \mathrm{m} \right) / \mathrm{s}$");
186 set(gcf, 'units', 'centimeters', 'position', [19, 1, 18, 15]);
187 grid on;
188 grid minor;
189 box on;
190 legend(legend_str, 'Location', 'Northwest');
191 hold off;
```

## D.6 Programa per evolució amb *Drag*

```

1  %% ROCKET LAUNCH WITH DRAG
2
3  clear all;
4  close all;
5  clc;
6
7  %% 1. Definition of Constants, Parameters and Variables
8
9  % 1.1. CONSTANTS
10
11  G = 6.67408e-11; % Gravitational Constant [m^3/(kg*s^2)]
12  R = 8.31432; % Universal Constant for Ideal Gases [J/mole*K]
13
14  % 1.1.1. Earth
15  R_Earth = 6371.0e3; % Radius of Earth [m]
16  M_Earth = 5.9724e24; % Earth's Mass [kg]
17  g0_Earth = 9.80665; % Acceleration at Earth's surface [m/s^2]
18  T0_Earth = 288.15; % US Standard Sea Level Temperature [K]
19  P0_Earth = 101325; % Pressure at Sea Level [Pa]
20  Mm_Earth = 28.9644*10^-3; % Molecular Mass [kg*mole^-1]
21  gamma_gas_Earth = 1.4; % Earth's air specific heats relation [adim]
22
23  % Earth's atmospheric layers altitude [m]
24  H_layer_Earth = 1e3*[0 11 20 32 47 52 61 69 79 90 100 110 117.776];
25  % Earth's atmospheric layers thermal gradient [K/m]
26  lambda_layer_Earth = 1e-3*[-6.5 0 1 2.8 0 -2 -4 -3 0 2 4.36 16.4596 0];
27
28  % 2.3. Compute base temperatures and pressures
29  [Tb, Pb] = getBaseTemperaturePressure(R, g0_Earth, T0_Earth, P0_Earth, Mm_Earth, H_layer_Earth, lambda_layer_Earth);
30
31  % 1. 3. VARIABLES
32  syms V; % Vehicle velocity [m/s]
33  syms gamma; % Vehicle flight path angle [rad]
34  syms h; % Vehicle altitude [m]
35  syms r; % Vehicle flight range [m]
36  syms t; % Vehicle time of flight [s]
37  syms g; % Acceleration [m/s^2]
38  syms Q; % Dynamic pressure at h [Pa]
39  syms W; % Vehicle weight [N]

```

```

40 syms C_D; % Drag Coefficient [adim]
41 syms A; % Vehicle reference area used in C_D [m^2]
42 %syms beta; % Ballistic coefficient [N/m^2]
43 syms C_L; % Vehicle lift coefficient [adim]
44
45
46 %% Input data and Previous calculation
47
48
49 S_ref = 25; % m^2
50 Pc = 100e5; % Pa
51 Tc = 3500; % K
52 gamma = 1.25;
53 m0 = 540e3; % kg
54 mW = 16e-3; % g/mol
55 m_dot = 2000; % kg/sec
56
57 Ae_At=[20;40;60;80];
58
59 R_comb = 8.31432/(mW);
60 c_car = sqrt(R_comb*Tc)/0.6581;
61 At = (m_dot*c_car)/Pc;
62 MPF_e(:,1) = 0.6581./Ae_At(:,1);
63
64 for i=1:length(MPF_e)
65     syms M;
66     M = vpasolve(MPF_e(i,1) == sqrt(gamma)* M / ((1 + (gamma-1)/2 *M^2)^(gamma+1)/(2*(gamma-1)))) , M, [1,6]);
67     Me(i,1) = M;
68 end
69
70 P_e = zeros(length(Ae_At),1);
71 for t=1:length(Ae_At)
72     P_e(t,1) = Pc/(1+Me(t,1)^2*((gamma-1)/2))^(gamma/(gamma-1));
73 end
74
75
76 %% Numerical integration
77
78 final_time = 120; % Final time for computations [s]
79 time_step = 0.5; % Step size [s]
80 time = 0:time_step:final_time; % Calculates upto final_time
81
82 h = zeros(length(Ae_At),length(time));

```

```

83 v = zeros(length(Ae_At),length(time));
84
85 h(:,1) = 0; % initial condition of altitude
86 v(:,1) = 0; % initial condition of velocity
87
88 % Altitude EDO
89
90 F_th = @(t,r,s) s;
91
92 % Velocity EDO
93
94 F_tv = @(t,r,s,p,q,c,m,ratio,coef) ...
95     (Pc*At*((2/(gamma+1))^(gamma+1))* (gamma*m+(1/m))/(sqrt(1+(m^2*(gamma-1)/2)))-((p/Pc)*ratio))-...
96     (0.5*q*s^2*S_ref*coef)/(m0-m_dot*t) - 9.80665; % Velocity EDO
97
98 for i=1:length(Ae_At)
99     for j=1:(length(time)-1) % calculation loop
100
101
102         [T, P, rho, a] = computeAtmosphere(Tb, Pb, H_layer_Earth, lambda_layer_Earth, R, g0_Earth, Mm_Earth, h(i,j));
103
104         Mach = v(i,j)/a;
105
106         if Mach ≤ 0.6
107
108             C_D = 0.15;
109
110             elseif Mach>0.6 && Mach ≤ 1.1
111
112                 C_D = -4.32*Mach^3 + 11.016*Mach^2 - 8.5536*Mach + 2.24952;
113
114             elseif Mach>1.1 && Mach ≤ 1.3
115
116                 C_D = -Mach^2 + 2.2*Mach - 0.79;
117
118             else
119
120                 C_D = 0.16769 + 0.17636/sqrt(Mach^2-1);
121
122             end
123
124             if P_e(i,1) < (0.4*P)

```



```

125 Me_prima(i,1) = sqrt((2/(gamma-1)) * ((Pc/(0.4*P))^((gamma-1)/gamma)-1));
126 Ae_At_prima(i,1) = 0.6581 / (sqrt(gamma)*Me_prima(i,1)/(1+((gamma-1)/2)*Me_prima(i,1)^2)^((gamma+1)/(2*(gamma-1))));
127
128
129 k_1h = F_th(time(j),h(i,j),v(i,j));
130 k_1v = F_tv(time(j),h(i,j),v(i,j),P,rho,a,Me_prima(i,1),Ae_At_prima(i,1),C_D);
131
132 k_2h = F_th(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v);
133 k_2v = F_tv(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v,P,rho,a,Me_prima(i,1),Ae_At_prima(i,1),C_D);
134
135 k_3h = F_th(time(j)+0.5*time_step,(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v));
136 k_3v = F_tv(time(j)+0.5*time_step,(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v),P,rho,a,Me_prima(i,1),...
137 Ae_At_prima(i,1),C_D);
138
139 k_4h = F_th(time(j)+time_step,(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step));
140 k_4v = F_tv(time(j)+time_step,(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step),P,rho,a,Me_prima(i,1),Ae_At_prima(i,1),C_D);
141
142
143 else
144
145 k_1h = F_th(time(j),h(i,j),v(i,j));
146 k_1v = F_tv(time(j),h(i,j),v(i,j),P,rho,a,Me(i,1),Ae_At(i,1),C_D);
147
148 k_2h = F_th(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v);
149 k_2v = F_tv(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v,P,rho,a,Me(i,1),Ae_At(i,1),C_D);
150
151 k_3h = F_th(time(j)+0.5*time_step,(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v));
152 k_3v = F_tv(time(j)+0.5*time_step,(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v),P,rho,a,Me(i,1),Ae_At(i,1),C_D);
153
154 k_4h = F_th(time(j)+time_step,(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step));
155 k_4v = F_tv(time(j)+time_step,(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step),P,rho,a,Me(i,1),Ae_At(i,1),C_D);
156
157 end
158
159 h(i,j+1) = h(i,j) + (1/6)*(k_1h+2*k_2h+2*k_3h+k_4h)*time_step; % main altitude equation
160 v(i,j+1) = v(i,j) + (1/6)*(k_1v+2*k_2v+2*k_3v+k_4v)*time_step; % main altitude equation
161
162 end
163
164 end
165
166 legend_str = cell(length(Ae_At), 1);
167 for i = 1:length(Ae_At)

```

```

168     legend_str(i,1) = {sprintf("$A_e/A_t = %d$", Ae_At(i))};
169     end
170
171     figure(1);
172     hold on;
173     title("\textbf{Altitude vs. Time (With Drag)}");
174     set(groot,'defaultAxesTickLabelInterpreter','latex');
175     set(groot,'defaulttextinterpreter','latex');
176     set(groot,'defaultLegendInterpreter','latex');
177     for i = 1:length(Ae_At)
178         plot(time, h(i,:)/1e3);
179     end
180     xlabel("Time $\left( \mathrm{s} \right)$");
181     ylabel("Altitude $\left( \mathrm{km} \right)$");
182     set(gcf, 'units', 'centimeters', 'position', [1, 1, 18, 15]);
183     grid on;
184     grid minor;
185     box on;
186     legend(legend_str, 'Location', 'Northwest');
187     hold off;
188
189
190     figure(2);
191     hold on;
192     title("\textbf{Velocity vs. Time (With Drag)}");
193     set(groot,'defaultAxesTickLabelInterpreter','latex');
194     set(groot,'defaulttextinterpreter','latex');
195     set(groot,'defaultLegendInterpreter','latex');
196     for i = 1:length(Ae_At)
197         plot(time, v(i,:));
198     end
199     xlabel("Time $\left( \mathrm{s} \right)$");
200     ylabel("Velocity $\left( \mathrm{m} \right) / \mathrm{s}$");
201     set(gcf, 'units', 'centimeters', 'position', [19, 1, 18, 15]);
202     grid on;
203     grid minor;
204     box on;
205     legend(legend_str, 'Location', 'Northwest');
206     hold off;

```

## D.7 Programa per evolució de forces

```

1  %% ROCKET LAUNCH WITH DRAG
2
3  clear all;
4  close all;
5  clc;
6
7
8  %% 1. Definition of Constants, Parameters and Variables
9
10 % 1.1. CONSTANTS
11
12 G = 6.67408e-11; % Gravitational Constant [m^3/(kg*s^2)]
13 R = 8.31432; % Universal Constant for Ideal Gases [J/mole*K]
14
15 % 1.1.1. Earth
16 R_Earth = 6371.0e3; % Radius of Earth [m]
17 M_Earth = 5.9724e24; % Earth's Mass [kg]
18 g0_Earth = 9.80665; % Acceleration at Earth's surface [m/s^2]
19 T0_Earth = 288.15; % US Standard Sea Level Temperature [K]
20 P0_Earth = 101325; % Pressure at Sea Level [Pa]
21 Mm_Earth = 28.9644*10^-3; % Molecular Mass [kg*mole^-1]
22 gamma_gas_Earth = 1.4; % Earth's air specific heats relation [adim]
23
24 % Earth's atmospheric layers altitude [m]
25 H_layer_Earth = 1e3*[0 11 20 32 47 52 61 69 79 90 100 110 117.776];
26 % Earth's atmospheric layers thermal gradient [K/m]
27 lambda_layer_Earth = 1e-3*[-6.5 0 1 2.8 0 -2 -4 -3 0 2 4.36 16.4596 0];
28
29 % 2.3. Compute base temperatures and pressures
30 [Tb, Pb] = getBaseTemperaturePressure(R, g0_Earth, T0_Earth, P0_Earth, Mm_Earth, H_layer_Earth, lambda_layer_Earth);
31
32 % 1. 3. VARIABLES
33 syms V; % Vehicle velocity [m/s]
34 syms gamma; % Vehicle flight path angle [rad]
35 syms h; % Vehicle altitude [m]
36 syms r; % Vehicle flight range [m]
37 syms t; % Vehicle time of flight [s]
38 syms g; % Acceleration [m/s^2]
39 syms Q; % Dynamic pressure at h [Pa]

```

```

40 syms W; % Vehicle weight [N]
41 syms C_D; % Drag Coefficient [adim]
42 syms A; % Vehicle reference area used in C_D [m^2]
43 %syms beta; % Ballistic coefficient [N/m^2]
44 syms C_L; % Vehicle lift coefficient [adim]
45
46
47 %% Input data and Previous calculation
48
49
50 S_ref = 25; % m^2
51 Pc = 100e5; % Pa
52 Tc = 3500; % K
53 gamma = 1.25;
54 m0 = 540e3; % kg
55 mW = 16e-3; % g/mol
56 m_dot = 2000; % kg/sec
57
58 Ae_At=[20;40;60;80];
59
60 R_comb = 8.31432/(mW);
61 c_car = sqrt(R_comb*Tc)/0.6581;
62 At = (m_dot*c_car)/Pc;
63 MPF_e(:,1) = 0.6581./Ae_At(:,1);
64
65 for i=1:length(MPF_e)
66     syms M;
67     M = vpasolve(MPF_e(i,1) == sqrt(gamma)* M / ((1 + (gamma-1)/2 *M^2)^(gamma+1)/(2*(gamma-1)))) , M, [1,6]);
68     Me(i,1) = M;
69 end
70
71 P_e = zeros(length(Ae_At),1);
72 for t=1:length(Ae_At)
73     P_e(t,1) = Pc/(1+Me(t,1)^2*((gamma-1)/2))^(gamma/(gamma-1));
74 end
75
76 %% Numerical integration
77
78
79 time_step=1; % step size
80 time = 0:time_step:122; % Calculates upto time (60)
81
82 h = zeros(length(Ae_At),length(time));

```

```

83 v = zeros(length(Ae_At),length(time));
84 thrust = zeros(length(Ae_At),length(time));
85 drag = zeros(length(Ae_At),length(time));
86 weight = zeros(length(Ae_At),length(time));
87
88 h(:,1) = 0; % initial condition of altitude
89 v(:,1) = 0; % initial condition of velocity
90
91 % Altitude EDO
92
93 F_th = @(t,r,s) s;
94
95 % Velocity EDO
96
97 F_tv = @(t,r,s,p,q,c,m,ratio,coef) ...
98     → (Pc*At*((2/(gamma+1))^(gamma+1)/(2*(gamma-1)))*(gamma*m+(1/m))/(sqrt(1+(m^2*(gamma-1)/2)))-(p/Pc)*ratio))-...
99     (0.5*q*s^2*S_ref*coef)/(m0-m_dot*t) - 9.80665; % Velocity EDO
100
101 for i=1:length(Ae_At)
102     % Calculation loop
103     for j=1:(length(time)-1)
104         [T, P, rho, a] = computeAtmosphere(Tb, Pb, H_layer_Earth, lambda_layer_Earth, R, g0_Earth, Mm_Earth, h(i,j));
105
106         Mach = v(i,j)/a;
107         if Mach ≤ 0.6
108             C_D = 0.15;
109         elseif Mach>0.6 && Mach ≤ 1.1
110             C_D = -4.32*Mach^3 + 11.016*Mach^2 - 8.5536*Mach + 2.24952;
111         elseif Mach>1.1 && Mach ≤ 1.3
112             C_D = -Mach^2 + 2.2*Mach - 0.79;
113         else
114             C_D = 0.16769 + 0.17636/sqrt(Mach^2-1);
115         end
116
117         if P_e(i,1) < (0.4*P)
118             Me_prima(i,1) = sqrt((2/(gamma-1)) * (Pc/(0.4*P))^(gamma-1)/gamma-1));
119             Ae_At_prima(i,1) = 0.6581 / (sqrt(gamma)*Me_prima(i,1)/(1+(gamma-1)/2)*Me_prima(i,1)^(gamma+1)/(2*(gamma-1))));
120
121             k_1h = F_th(time(j),h(i,j),v(i,j));
122             k_1v = F_tv(time(j),h(i,j),v(i,j),P,rho,a,Me_prima(i,1),Ae_At_prima(i,1),C_D);
123
124             k_2h = F_th(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v);
125             k_2v = F_tv(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v,P,rho,a,Me_prima(i,1),Ae_At_prima(i,1),C_D);

```

```

125
126 k_3h = F_th((time(j)+0.5*time_step), (h(i,j)+0.5*time_step*k_2h), (v(i,j)+0.5*time_step*k_2v));
127 k_3v = F_tv((time(j)+0.5*time_step), (h(i,j)+0.5*time_step*k_2h), (v(i,j)+0.5*time_step*k_2v), P, rho, a, Me_prima(i,1), ...
128 Ae_At_prima(i,1), C_D);
129
130 k_4h = F_th((time(j)+time_step), (h(i,j)+k_3h*time_step), (v(i,j)+k_3v*time_step));
131 k_4v = F_tv((time(j)+time_step), (h(i,j)+k_3h*time_step), (v(i,j)+k_3v*time_step), P, rho, a, Me_prima(i,1), Ae_At_prima(i,1), C_D);
132
133 thrust(i,j) = (Pc*At*((2/(gamma+1))^(gamma+1)/(2*(gamma-1))))*(gamma*Me_prima(i,1)+(1/Me_prima(i,1)))/...
134 (sqrt(1+(Me_prima(i,1)^2*(gamma-1)/2))-((P/Pc)*Ae_At_prima(i,1)));
135
136
137 k_1h = F_th(time(j),h(i,j),v(i,j));
138 k_1v = F_tv(time(j),h(i,j),v(i,j),P,rho,a,Me(i,1),Ae_At(i,1),C_D);
139
140 k_2h = F_th(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v);
141 k_2v = F_tv(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v,P,rho,a,Me(i,1),Ae_At(i,1),C_D);
142
143 k_3h = F_th((time(j)+0.5*time_step), (h(i,j)+0.5*time_step*k_2h), (v(i,j)+0.5*time_step*k_2v));
144 k_3v = F_tv((time(j)+0.5*time_step), (h(i,j)+0.5*time_step*k_2h), (v(i,j)+0.5*time_step*k_2v), P, rho, a, Me(i,1), Ae_At(i,1), C_D);
145
146 k_4h = F_th((time(j)+time_step), (h(i,j)+k_3h*time_step), (v(i,j)+k_3v*time_step));
147 k_4v = F_tv((time(j)+time_step), (h(i,j)+k_3h*time_step), (v(i,j)+k_3v*time_step), P, rho, a, Me(i,1), Ae_At(i,1), C_D);
148
149 thrust(i,j) = (Pc*At*((2/(gamma+1))^(gamma+1)/(2*(gamma-1))))*(gamma*Me(i,1)+(1/Me(i,1)))/(sqrt(1+(Me(i,1)^2*(gamma-1)/2)))-...
150 ((P/Pc)*Ae_At(i,1)));
151
152
153 h(i,j+1) = h(i,j) + (1/6)*(k_1h+2*k_2h+2*k_3h+k_4h)*time_step; % main altitude equation
154 v(i,j+1) = v(i,j) + (1/6)*(k_1v+2*k_2v+2*k_3v+k_4v)*time_step; % main altitude equation
155
156 drag(i,j) = 0.5*rho*v(i,j)^2*S_ref*C_D;
157 weight(i,j) = (m0-m_dot*time(j))*g0_Earth;
158
159 end
160
161 figure(1);
162 hold on;
163 title("\textbf{Forces vs. Time ($A_e/A_t = 20$)}");
164 set(groot,'defaultAxesFickLabelInterpreter','latex');
165 set(groot,'defaulttextinterpreter','latex');
166 set(groot,'defaultLegendInterpreter','latex');
167 plot(time, weight(1,:)/1e3, 'r');

```

```

168 plot(time, drag(1,:)/le3, 'g');
169 plot(time, thrust(1,:)/le3, 'b');
170 xlabel('Time $\left( \mathrm{s} \right)$ \right)$');
171 ylabel('Force $\left( \mathrm{kN} \right)$ \right)$');
172 xlim([0 120]);
173 set(gcf, 'units', 'centimeters', 'position', [0, 1, 18, 15]);
174 grid on;
175 grid minor;
176 box on;
177 legend('Weight', 'Drag', 'Thrust', 'Location', 'Northwest');
178 hold off;
179
180 figure(2);
181 hold on;
182 title("\textbf{Forces vs. Time ($A_e/A_t = 40$)}");
183 set(groot, 'defaultAxesTickLabelInterpreter', 'latex');
184 set(groot, 'defaultTextInterpreter', 'latex');
185 set(groot, 'defaultLegendInterpreter', 'latex');
186 plot(time, weight(2,:)/le3, 'r');
187 plot(time, drag(2,:)/le3, 'g');
188 plot(time, thrust(2,:)/le3, 'b');
189 xlabel('Time $\left( \mathrm{s} \right)$ \right)$');
190 ylabel('Force $\left( \mathrm{kN} \right)$ \right)$');
191 xlim([0 120]);
192 set(gcf, 'units', 'centimeters', 'position', [0, 1, 18, 15]);
193 grid on;
194 grid minor;
195 box on;
196 legend('Weight', 'Drag', 'Thrust', 'Location', 'Northwest');
197 hold off;
198
199
200 figure(3);
201 hold on;
202 title("\textbf{Forces vs. Time ($A_e/A_t = 60$)}");
203 set(groot, 'defaultAxesTickLabelInterpreter', 'latex');
204 set(groot, 'defaultTextInterpreter', 'latex');
205 set(groot, 'defaultLegendInterpreter', 'latex');
206 plot(time, weight(3,:)/le3, 'r');
207 plot(time, drag(3,:)/le3, 'g');
208 plot(time, thrust(3,:)/le3, 'b');
209 xlabel('Time $\left( \mathrm{s} \right)$ \right)$');
210 ylabel('Force $\left( \mathrm{kN} \right)$ \right)$');

```

```

211 xlim([0 120]);
212 set(gcf, 'units', 'centimeters', 'position', [0, 1, 18, 15]);
213 grid on;
214 grid minor;
215 box on;
216 legend('Weight', 'Drag', 'Thrust', 'Location', 'Northwest');
217 hold off;
218
219
220 figure(4);
221 hold on;
222 title("\textbf{Forces vs. Time ($A_e/A_t = 80$)}");
223 set(groot, 'defaultAxisTickLabelInterpreter', 'latex');
224 set(groot, 'defaultTextInterpreter', 'latex');
225 set(groot, 'defaultLegendInterpreter', 'latex');
226 plot(time, weight(4,:)/1e3, 'r');
227 plot(time, drag(4,:)/1e3, 'g');
228 plot(time, thrust(4,:)/1e3, 'b');
229 xlabel('Time $\left( \mathrm{s} \right)$');
230 ylabel('Force $\left( \mathrm{kN} \right)$');
231 xlim([0 120]);
232 set(gcf, 'units', 'centimeters', 'position', [0, 1, 18, 15]);
233 grid on;
234 grid minor;
235 box on;
236 legend('Weight', 'Drag', 'Thrust', 'Location', 'Northwest');
237 hold off;
238
239 fprintf("%15s%15s%15s\n", "Ae/At", "Thrust (N)", "Drag (N)");
240 for i = 1:length(Ae_At)
241     fprintf("%15d", Ae_At(i));
242     fprintf("%15.3f", max(thrust(i,:)));
243     fprintf("%15.3f\n", max(drag(i,:)));
244 end
245
246
247 fprintf("%15s%15s%15s%15s\n", "Ae/At", "Thrust (kN)", "Isp (s)", "Drag (kN)");
248 for i = 1:length(Ae_At)
249     fprintf("%15d&", Ae_At(i));
250     fprintf("%15.0f&", max(thrust(i,:))/1e3);
251     fprintf("%15.0f&", max(thrust(i,:))/(g0_Earth*m_dot));
252     fprintf("%15.0f \\\n", max(drag(i,:))/1e3);
253 end

```



## D.8 Programa per càlcul de l'alçada i velocitat final

```

1  %% ROCKET LAUNCH
2
3  clear all;
4  close all;
5  clc;
6
7  %% 1. Definition of Constants, Parameters and Variables
8
9  % 1.1. CONSTANTS
10
11  G = 6.67408e-11; % Gravitational Constant [m^3/(kg*s^2)]
12  R = 8.31432; % Universal Constant for Ideal Gases [J/mole*K]
13
14  % 1.1.1. Earth
15  R_Earth = 6371.0e3; % Radius of Earth [m]
16  M_Earth = 5.9724e24; % Earth's Mass [kg]
17  g0_Earth = 9.80665; % Acceleration at Earth's surface [m/s^2]
18  T0_Earth = 288.15; % US Standard Sea Level Temperature [K]
19  P0_Earth = 101325; % Pressure at Sea Level [Pa]
20  Mm_Earth = 28.9644*10^-3; % Molecular Mass [kg*mole^-1]
21  gamma_gas_Earth = 1.4; % Earth's air specific heats relation [adim]
22
23  % Earth's atmospheric layers altitude [m]
24  H_layer_Earth = 1e3*[0 11 20 32 47 52 61 69 79 90 100 110 117.776];
25  % Earth's atmospheric layers thermal gradient [K/m]
26  lambda_layer_Earth = 1e-3*[-6.5 0 1 2.8 0 -2 -4 -3 0 2 4.36 16.4596 0];
27
28  % 2.3. Compute base temperatures and pressures
29  [Tb, Pb] = getBaseTemperaturePressure(R, g0_Earth, T0_Earth, P0_Earth, Mm_Earth, H_layer_Earth, lambda_layer_Earth);
30
31
32  % 1. 3. VARIABLES
33  syms V; % Vehicle velocity [m/s]
34  syms gamma; % Vehicle flight path angle [rad]
35  syms h; % Vehicle altitude [m]
36  syms r; % Vehicle flight range [m]
37  syms t; % Vehicle time of flight [s]
38  syms g; % Acceleration [m/s^2]
39  syms Q; % Dynamic pressure at h [Pa]

```

```

40 syms W; % Vehicle weight [N]
41 syms C_D; % Drag Coefficient [adim]
42 syms A; % Vehicle reference area used in C_D [m^2]
43 %syms beta; % Ballistic coefficient [N/m^2]
44 syms C_L; % Vehicle lift coefficient [adim]
45
46
47
48 %% Input data and Previous calculation
49
50 Pc = 100 * P0_Earth; % Pa
51 Tc = 3500; % K
52 gamma = 1.25;
53 m0 = 540e3; % kg
54 mW = 16e-3; % kg/mol
55 m_dot = 2000; % kg/sec
56 Ae_At = (2:0.5:80)';
57 R_comb = R/(mW);
58 c_car = sqrt(R_comb*Tc)/0.6581; % 0.6581 from tables
59 At = (m_dot*c_car)/Pc;
60 MPF_e(:,1) = 0.6581./Ae_At(:,1);
61
62 % Solving MFP equation to obtain Mach Number
63 Me = zeros(length(MPF_e),1);
64 for i=1:length(MPF_e)
65     syms M;
66     M = vpasolve(MPF_e(i,1) == sqrt(gamma)* M / ((1 + (gamma-1)/2 *M^2)^(gamma+1)/(2*(gamma-1)))) , M, [1,6]);
67     Me(i,1) = M;
68 end
69
70 P_e = zeros(length(Ae_At),1);
71 for t=1:length(Ae_At)
72     P_e(t,1) = Pc/(1+Me(t,1)^2*((gamma-1)/2)^(gamma/(gamma-1)));
73 end
74
75 %% Numerical integration
76
77 final_time = 120; % Final time for computations [s]
78 time_step = 0.5; % Step size [s]
79 time = 0:time_step:final_time; % Calculates upto final_time
80
81 h = zeros(length(Ae_At),length(time));
82 v = zeros(length(Ae_At),length(time));

```

```

83
84 h(:,1) = 0; % initial condition of altitude
85 v(:,1) = 0; % initial condition of velocity
86
87 F_th = @ (t,r,s) s; % Altitude EDO
88
89 F_tv = @ (t,r,s,p,q,c,m,ratio) ...
    ↳ (Pc*At*((2/(gamma+1))^(gamma+1)/(2*(gamma-1))))*(gamma*m+(1/m))/(sqrt(1+(m^2*(gamma-1)/2)))-((p/Pc)*ratio))/(m0-m_dot*t)) - ...
    ↳ g0_Earth; % Velocity EDO
90
91 for i=1:length(Ae_At)
92
93     for j=1:(length(time)-1) % calculation loop
94
95         [T, P, rho, a] = computeAtmosphere(Tb, Pb, H_layer_Earth, lambda_layer_Earth, R, g0_Earth, Mm_Earth, h(i,j));
96
97         if P_e(i,1) < (0.4*P)
98
99             Me_prima(i,1) = sqrt((2/(gamma-1)) * ((Pc/(0.4*P))^(gamma-1)/gamma)-1));
100             Ae_At_prima(i,1) = 0.6581 / (sqrt(gamma)*Me_prima(i,1)/(1+(gamma-1)/2)*Me_prima(i,1)^2)^((gamma+1)/(2*(gamma-1))));
101
102             k_1h = F_th(time(j),h(i,j),v(i,j));
103             k_1v = F_tv(time(j),h(i,j),v(i,j),P,rho,a,Me_prima(i,1),Ae_At_prima(i,1));
104
105             k_2h = F_th(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v);
106             k_2v = F_tv(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v,P,rho,a,Me_prima(i,1),Ae_At_prima(i,1));
107
108             k_3h = F_th((time(j)+0.5*time_step),(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v));
109             k_3v = F_tv((time(j)+0.5*time_step),(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v),P,rho,a,Me_prima(i,1),Ae_At_prima(i,1));
110             Ae_At_prima(i,1);
111
112             k_4h = F_th((time(j)+time_step),(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step));
113             k_4v = F_tv((time(j)+time_step),(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step),P,rho,a,Me_prima(i,1),Ae_At_prima(i,1));
114
115             %Cf_v (i,1) = ((2/(gamma+1))^(gamma+1)/(2*(gamma-1)))* (gamma*Me(i,1)+(1/Me(i,1)))/(sqrt(1+(Me(i,1)^2*(gamma-1)/2)));
116
117             else
118
119             k_1h = F_th(time(j),h(i,j),v(i,j));
120             k_1v = F_tv(time(j),h(i,j),v(i,j),P,rho,a,Me(i,1),Ae_At(i,1));
121
122             k_2h = F_th(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v);
123

```

```

124 k_2v = F_tv(time(j)+0.5*time_step,h(i,j)+0.5*time_step*k_1h,v(i,j)+0.5*time_step*k_1v,P,rho,a,Me(i,1),Ae_At(i,1));
125
126 k_3h = F_th((time(j)+0.5*time_step),(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v));
127 k_3v = F_tv((time(j)+0.5*time_step),(h(i,j)+0.5*time_step*k_2h),(v(i,j)+0.5*time_step*k_2v),P,rho,a,Me(i,1),Ae_At(i,1));
128
129 k_4h = F_th((time(j)+time_step),(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step));
130 k_4v = F_tv((time(j)+time_step),(h(i,j)+k_3h*time_step),(v(i,j)+k_3v*time_step),P,rho,a,Me(i,1),Ae_At(i,1));
131
132 end
133
134 h(i,j+1) = h(i,j) + (1/6)*(k_1h+2*k_2h+2*k_3h+k_4h)*time_step; % main altitude equation
135 v(i,j+1) = v(i,j) + (1/6)*(k_1v+2*k_2v+2*k_3v+k_4v)*time_step; % main altitude equation
136
137 end
138
139 end
140
141
142 figure(1);
143 hold on;
144 str = sprintf("\textbf{Final altitude vs. $A_e / A_t$ at $t_b = %d$ \\\mathrm{s}$ (Without drag)}", final_time);
145 title(str);
146 set(groot,'defaultAxesTickLabelInterpreter','latex');
147 set(groot,'defaulttextinterpreter','latex');
148 set(groot,'defaultLegendInterpreter','latex');
149 plot(Ae_At, h(:,end)/1e3, 'b');
150 xlabel("$A_e / A_t$");
151 ylabel("Altitude $\left( \mathrm{km} \right)$");
152 set(gcf, 'units', 'centimeters', 'position', [1, 1, 18, 15]);
153 grid on;
154 grid minor;
155 box on;
156 hold off;
157
158 figure(2);
159 hold on;
160 str = sprintf("\textbf{Final velocity vs. $A_e / A_t$ at $t_b = %d$ \\\mathrm{s}$ (Without drag)}", final_time);
161 title(str);
162 set(groot,'defaultAxesTickLabelInterpreter','latex');
163 set(groot,'defaulttextinterpreter','latex');
164 set(groot,'defaultLegendInterpreter','latex');
165 plot(Ae_At, v(:,end), 'b');
166 xlabel("$A_e / A_t$");

```

```
167 ylabel("Velocity $\left( \text{ m / s } \right)$");
168 set(gcf, 'units', 'centimeters', 'position', [19, 1, 18, 15]);
169 grid on;
170 grid minor;
171 box on;
172 hold off;
```

## Referències

- [1] Niklas Andersson. *High-speed flows*. 2016. URL: <https://nikander.github.io/g3dflow/Stories/HighSpeedFlows/> (cons. 19-06-2020).
- [2] Carles Bonet Reves et al. «Càlcul numèric». A: 2a ed. Edicions UPC, 2012. Cap. 9, pàg. 229, 238 - 241.
- [3] Josep Oriol Lizandra Dalmases. «Mass Flow Parameter». Document PDF. Terrassa, 2019.